

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

NASA TECHNICAL MEMORANDUM

NASA TM-78290

SCHEDULER SOFTWARE FOR TRACKING AND DATA RELAY SATELLITE SYSTEM LOADING ANALYSIS - USER MANUAL AND PROGRAMMER GUIDE

By Ray Craft, Christel Dunn, Janet McCord,
and Louis Simeone
Data Systems Laboratory

June 13, 1980

(NASA-TM-78290) SCHEDULER SOFTWARE FOR
TRACKING AND DATA RELAY SATELLITE SYSTEM
LOADING ANALYSIS: USER MANUAL AND
PROGRAMMER GUIDE (NASA) 101 p HC A06/MF A01

N80-31070

CSCL 09B G3/61

Unclas
28540

NASA

*George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama*

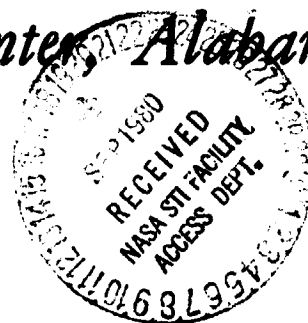


TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. LANDSAT SCHEDULER	5
2.1 General	6
2.2 Interactive Capabilities	7
2.3 Program Flow.....	11
2.4 Subroutines	20
3. TDRS SCHEDULER	31
3.1 General	32
3.2 Interactive Capabilities	34
3.3 Program Flow	38
3.4 Subroutines	42
APPENDIX A - LANDSAT SCHEDULER	63
APPENDIX B - TDRS SCHEDULER	72
APPENDIX C - FILESET	86
APPENDIX D - TLGEN	92

PRECEDING PAGE BLANK NOT FILMED

LIST OF ILLUSTRATIONS

Figure	Title	Page
1-1.	TDRS loading analysis	3
2-1.	Landsat Scheduler input setup parameters	8
2-2.	Landsat Scheduler functional block diagram	12
2-3.	Landsat Scheduler sample scheduling diagram for Landsat D	18
3-1.	TDRS Scheduler functional block diagram	39

LIST OF TABLES

Table	Title	Page
3-1.	Time Line Formats (32 Words)	33
3-2.	TDRS Scheduler Resource and Requestor Setup Files	35

1. INTRODUCTION

1. INTRODUCTION

The Software Engineering Division, in coordination with the Requirements and Analysis Division and the Engineering Computers Division, designed and implemented a Fortran computer program which models and simulates the data relay activities for a system of satellite payloads and Tracking Data Relay Satellites (TDRS). Although the immediate goal was to simulate a specific activity, the software was designed with some consideration to broader applications. This is particularly true of the TDRS scheduler. In general, a set of available resources and potential resource users (requestors to be serviced) are defined by input parameters. Disk files are specified (after preparation by auxiliary software) which contain resource available time lines and requestor/resource compatibility time lines, i.e., line-of-sight (LOS) time lines when scheduling satellite to TDRS relay activities. Requests are generated, based upon any of several models selected, in the form of request windows which specify the time period in which a resource can be used. The request window may be equal to or greater than the time period required to satisfy the request. A request will not be scheduled unless an additional constraint, imposed by a time line file, is met, e.g., satellite to TDRS LOS. Requests are scheduled on a priority basis and by order of entry into the requestor array. Resource available windows become candidates for scheduling by chronological order and by order of entry. All scheduling activities are recorded as events in a disk file which is available for further analyses or entry into subsequent scheduling runs. Various statistical data are logged during the program execution for printout to the user. In addition to these general capabilities, the software is highly modular to facilitate modifications. For example, new request models or different schemes for dynamically assigning priority can be easily accommodated.

Currently, the software is operational on the PRIME 400 mini-computer in Data Systems Laboratory. Many small utility programs may be used to facilitate specific applications, but there are two primary programs - Landsat Scheduler and TDRS Scheduler.

Landsat Scheduler schedules the real-time (R/T) data relays from the orbiting Landsat D and Landsat D' payloads through two geostationary TDRS's. TDRS Scheduler primarily schedules tape recorder (T/R) dumps from the remaining satellite payloads through a system of TDRS's. The TDRS Scheduler also has the capability for real-time data relay scheduling, described in Section 3.2 under Interactive Capabilities. Figure 1-1 is a block diagram of the entire system, showing the relationship of these programs to the DSDS software. The DSDS is used to generate LOS time lines between the payloads and the TDRS's, reflecting time of acquisition and loss of sight windows.

Each program allows operator interaction for setup through a series of prompts and responses at a PRIME 400 computer CRT terminal, described in detail for each program under Interactive Capabilities.

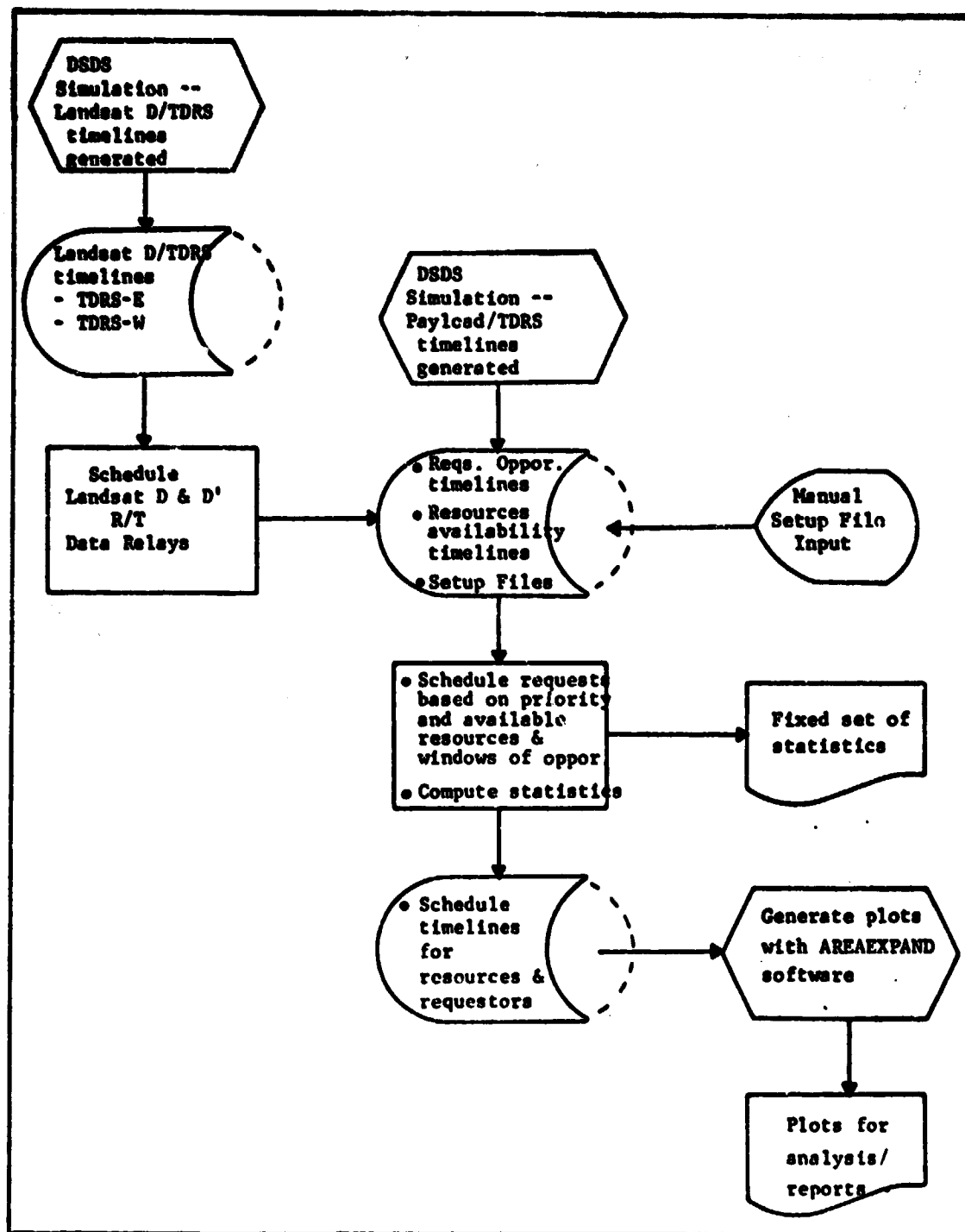


Figure 1-1. TDRS loading analysis.

The operator must enter setup, as well as run-time parameters. The TDRS Scheduler allows the operator to do a program interrupt/restart to change some of these parameters with different setup files. In addition the Scheduler allows the operator to make iterative program runs to use the remaining available TDRS resource windows from the previous run for scheduling other requestor requirements.

To obtain an ASCII printout of the binary output files, the operator uses the program #NEW.OUTPUT. Finally the existing interactive plot software called AREAEXPAND is used to generate various plots from the output files for detailed analysis and reports.

2. LANDSAT SCHEDULER

2.1 GENERAL

2.1.1 Landsat D & D' Program

The Landsat Scheduler program LAND.D&DP schedules real-time (R/T) data relays from Landsat D and D' satellites through two geostationary TDRS's - TDRS East and TDRS West. Each TDRS is scheduled by separate program execution runs using unique setup parameters for each.

The LAND.D&DP program uses, as input, either the output line-of-sight (LOS) time line files generated by the DSDS simulation software program, or the output time line files generated by the TLGEN program. (See Appendix D for TLGEN program description and operating instructions.)

Basically, the LAND.D&DP program reads an "LOS window" (time of TDRS acquisition and loss of sight) from the input file and attempts to schedule a "request window" (a time period when transmission of data occurs) for odd and even orbits, both Landsat D and D', according to the setup parameters established by the user at the beginning of the program. Since TDRS East and TDRS West are run as separate program executions, provisions should be made in the setup parameters to allow, for a given payload, only one TDRS to be receiving data during a given time period. This can be accomplished by setting the TDRS East's potential scheduling start time to zero and ending with a length of one-half of the orbital period; for TDRS West, potential scheduling should start at one-half the orbital period and end with a length of one-half of the orbital period (see Section 2.2.1). The detailed scheduling algorithm is described in Section 2.3.

The LAND.D&DP also has the capability of reserving specific time periods for hardware preventative maintenance (PM), when neither TDRS East nor TDRS West will be allowed to receive/transmit data. This capability is also established at the beginning of the program when the user inputs the setup parameters.

Scheduled "request windows" (32-word records containing the "on-off" time periods and related scheduling data) are written to an output disk file in the same format as the input file (see Figure 4-1 for input/output file layout). The LAND.D&DP output file will be used as an input file for the TDRS Scheduler.

2.1.2 Landsat D Program

The Landsat Scheduler program LAND.D is identical to LAND.D&DP, except the LAND.D program does not schedule a Landsat D' odd or even orbit.

2.2 INTERACTIVE CAPABILITIES

2.2.1 General

Before executing the Landsat Scheduler, the user should be instructed as to which program to use (see Section 2.1).

Particular attention should also be given to proper setup of the parameters DLTAT1 and DLTAT2, according to the TDRS that is being processed. Generally, the following will apply:

	<u>TDRS East</u>	<u>TDRS West</u>
DLTAT1 - Start of Potential Scheduling Period (1 min)	0.	TORB/2.
DLTAT2 - Length of Potential Scheduling Period (in min)	TORB/2.	TORB/2.

This will prevent transmission of data occurring simultaneously to TDRS East and TDRS West from the same payload.

2.2.2 Operator Instructions

The initialization and run-time procedures for the Landsat Scheduler are as follows:

1. Sign in on the PRIME 400 using the appropriate user file directory (UFD) where the files to be used are located.
2. For a hard copy of everything that appears on the CRT during a program execution run, enter 'COMO filename' for eventual spooling to the line printer.
3. To initiate program execution, enter one of the following:

'SEG #LAND.D&DP'	(for the Landsat D & D' Program)
or	
'SEG #LAND.D'	(for the Landsat D Program)
4. Enter the setup parameters by responding to the CRT prompts, as specified in Figure 2-1.
5. The Landsat Scheduler will print the scheduled results to the CRT. Normal termination of scheduling will result in one of the following messages to be printed at the end of the scheduling data:

'END OF FILE RFLG = 1'
or
'BEYOND MISSION STOP TIME RFLG = 2'

<u>SET PROMPT MESSAGE/DESCRIPTION</u>	<u>FORMAT</u>	<u>TYPE</u>	<u>MAX. CHARS.</u>	<u>REMARKS</u>
1. Enter Current Run Date and Time	5%	Integer	4 ss.	Self-explanatory. (NOTE: A space or comma between each of the fields will be treated as a delimiting mark.) Ex.: MM,DD,YY,HH,MM
2. Enter Operators Name	6A4	Alpha-Num	24	Self-explanatory.
3. Enter INAME -- Input File Name	4A2	Alpha-Num	6	Six-character input file name.
4. Enter ONAME -- Output File Name	4A2	Alpha-Num	6	Six-character output file name.
5. Enter REQID(1) -- Request ID for Landsat D	I4	Integer	4	ID code to represent Landsat D.
6. Enter REQID(2) -- Request ID for Landsat D'	I4	Integer	4	ID code to represent Landsat D'.
7. Enter REQID(3) -- Request ID for FM Time	I4	Integer	4	ID code to represent FM time.
8. Enter RESID -- Resource ID Code	I4	Integer	4	ID code to represent the type of resource (i.e., TDRS-E, TDRS-W).
9. Enter REQTYPE -- Request Type Code	I4	Integer	4	ID code to represent the type of request (i.e., R/T Relay, T/R Dump).
10. Enter MSTART -- Mission Start Time (in Mins)	F15.6	Real	15	Self-explanatory.
11. Enter MSTOP -- Mission Stop Time (in Mins)	F15.6	Real	15	Self-explanatory.
12. Enter TORB -- Orbital Period in Mins	F15.6	Real	15	Length of each orbital period.
13. Enter DTIME -- Duration Time of R/T Relay (in Mins)	F15.6	Real	15	Length of each R/T relay.
14. Enter TOFFST -- Offset Time Between Landsat D & D' in Mins	F15.6	Real	15	Distance between Landsat D & D' in Mins.
15. Enter DLAT1 -- Start of Potential Scheduling Period in Mins	F15.6	Real	15	Self-explanatory.
16. Enter DLAT2 -- Length of Potential Scheduling Period in Mins	F15.6	Real	15	Self-explanatory.
17. Enter PMSTRT -- Start Time of the First FM Period in Mins	F15.6	Real	15	First Preventative Maintenance start time.
18. Enter PMDUR -- Duration Time of Each FM Period in Mins	F15.6	Real	15	Length of each FM.
19. Enter PMFRQ -- Time Between Successive FM Periods in Mins	F15.6	Real	15	Frequency of each FM.

Figure 2-1. Landsat Scheduler input setup parameters.

(Other program messages printed to the CRT during execution are described in Section 2.2.3. Error messages and their corrective action are explained in Appendix A.3.)

Following the scheduling results, the program will print a statistical summary and then terminate. (See sample printout of the program output in Appendix 4.1.)

6. If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. To spool the COMO file and get a hard copy at the line printer, enter 'SPOOL filename -FTN'.

7. To convert the binary output disk file to a readable ASCII format, enter 'SEG #NEW.OUTPUT'. Respond to the CRT prompt for the output filename by entering either a spooling filename for an eventual ASCII print to the line printer, or a carriage return for an ASCII print to the CRT. (See sample printout of the output disk file in Appendix A.2.)

8. If desired, the operator may run the AREAEXPAND program to generate various plots for detailed analyses and reports. (See sample printout of the AREAEXPAND output in Appendix B.5.)

2.2.3 Program Messages

Listed below are the possible program messages that may appear while executing the Landsat Scheduler programs. Following each message will be the name of the subroutine supplying the message and a brief explanation, where needed. (NOTE: Messages supplied by DPRIME will never appear in the LAND.D program since Landsat D' is never scheduled.)

'***** CANT SCHEDULE ODD-NO ORBIT XXXXX. *****'
(LANDSA - Self-explanatory)

'***** CANT SCHEDULE EVEN-NO ORBIT XXXXX. *****'
(LANDSA - Self-explanatory)

'***** ORBNO XXXXX. - (T4,T3) ADJUSTED BY XXXXX.XXX
MINUTES *****'
(LANDSA - In order to fit within the "partial
IOS window," the "request window" was
adjusted to the left.)

'***** ORBNO XXXXX. - (T3,T4) ADJUSTED BY XXXXX.XXX
MINUTES *****'
(LANDSA - In order to fit within the "partial
IOS window," the "request window" was
adjusted to the right.)

'***** LANDSAT D NOT SCHEDULED *****'

(DPRIME - If Landsat D was not scheduled,
Landsat D' will not be scheduled.)

'***** BEYOND STOP TIME *****'

(DPRIME - Exceeded mission stop time while
trying to schedule Landsat D'.)

'***** RESERVED FOR PM TIME *****'

(DPRIME - Could not schedule "requested
window" because it occurred within a preventative
maintenance period.)

'END OF FILE RFLG = 1'

(DREAD - End of disk input file encountered;
normal termination.)

'BEYOND MISSION STOP TIME RFLG = 2'

(DREAD - "Request window" has exceeded
mission stop time; normal termination.)

'REQUESTED-STOP-TIME BEYOND MISSION-STOP-TIME, SO
MISSION-STOP-TIME USED'

(DREAD - Self-explanatory.)

2.3 PROGRAM FLOW

2.3.1 General

Figure 2-2 is a block diagram of the general data flow for the Landsat Scheduler programs. The complete diagram represents the LAND.D&DP program. LAND.D is identical to LAND.D&DP except for the portion shown within the dashed lines.

The LAND.D&DP algorithm will be described in 2.3.2 and will indicate the exceptions for LAND.D, when applicable.

The following general information may be helpful in understanding the detailed scheduling algorithm.

1. A pair of records, as read from the input file, are needed to represent an "LOS window" - one being a beginning-LOS record ("on") and the other being an ending-LOS record ("off"). (For the input file to be valid, it must contain records that have alternating "on-off" status codes.) The on-off codes designate when a Landsat can be seen by the TDRS. The programs translate an "on" code to mean "window on" (WON) and an "off" code to mean "window off" (WOFF). WON will be defined as being greater than or equal to MSTART, and WOFF will be defined as being less than or equal to MSTOP.

2. Some Landsat-TDRS contacts can last for more than one Landsat orbit. In order to process these, the program limits the line-of-sight windows to a single orbital period (TORB) by using the "partial window on" (TON) and "partial window off" (TOFF) parameters. While WON and WOFF are used to hold the window on-off pair as read from the input disk file, TON and TOFF are used to hold the partial window on-off pair for a current orbit being scheduled. TON will be defined as being greater than or equal to the beginning orbit time (BEGORB), and TOFF will be defined as being less than or equal to the ending orbit time (ENDORB).

3. The orbit number (ORBNO) is the current number of the orbit. Orbit number one starts at MSTART time. As each orbit begins, the orbit number is incremented, and the beginning orbit (BEGORB) and ending orbit (ENDORB) times are recomputed as follows:

$$\text{BEGORB} = \text{MSTART} + \text{TORB}(\text{ORBNO} - 1.0) + \text{DLTAT1}$$

$$\text{ENDORB} = \text{BEGORB} + \text{DLTAT2}$$

4. As stated earlier, preventative maintenance (PM) periods are time periods ("PM windows") that will be reserved for hardware preventative maintenance. No TDRS-payload relays will be allowed during these time periods. A PM window consists of a beginning and ending time period that is computed as follows:

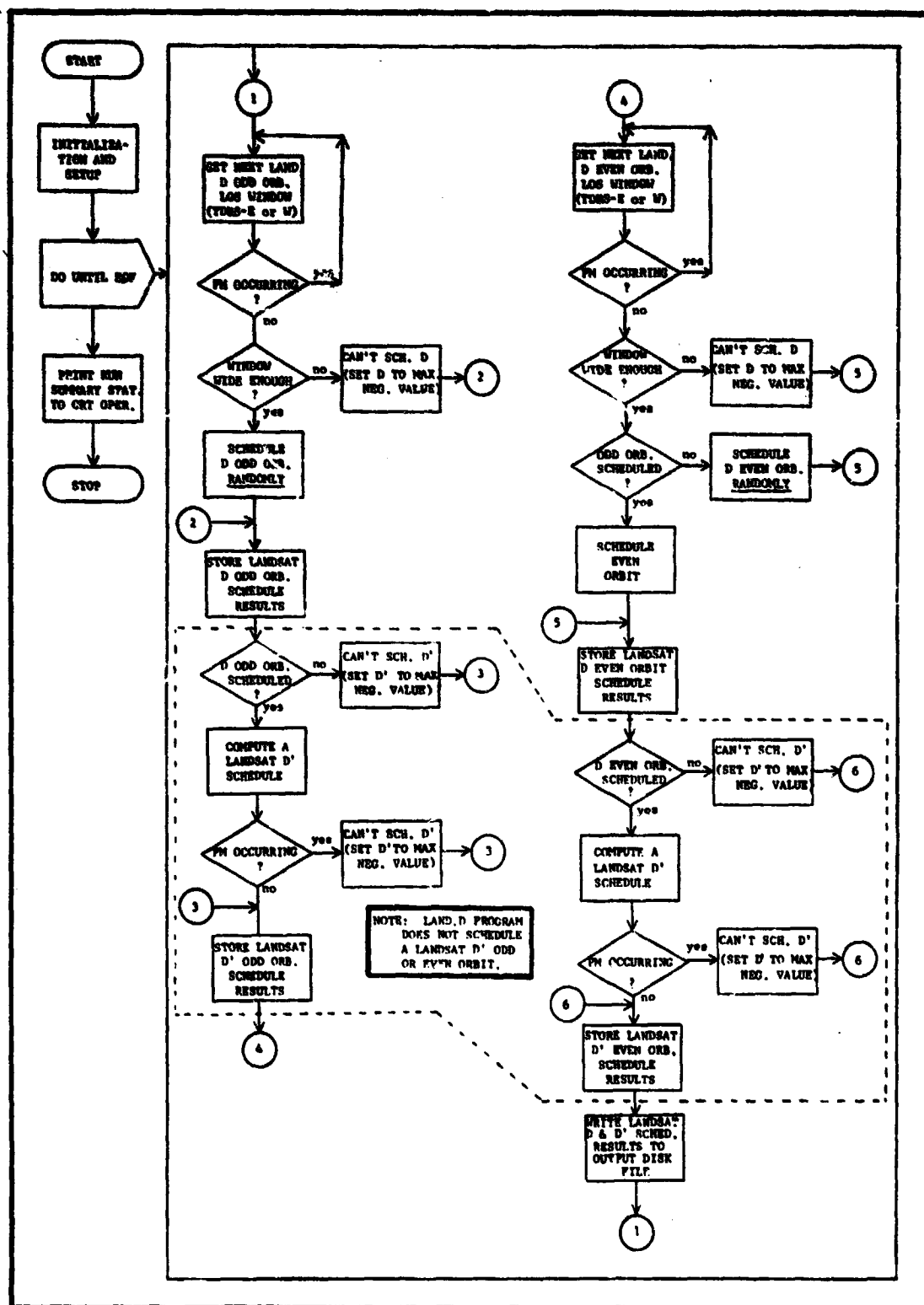


Figure 2-2. Landsat Scheduler program block diagram.

The Very
First Window

All
Successive
Windows

PM Start Time (PMT1) = MSTART + PMSTRT

PMT1 + PMSTRT

PM Stop Time (PMT2) = PMT1 + PMDUR

PMT1 + PMDUR

A complete list of the program parameters and definitions can be found in Section 2.4.

2.3.2 Scheduling Algorithm

Each execution of one of the Landsat Scheduler programs is designed to process the receipt/transmission of R/T data relays for a single TDRS. This method of scheduling allows maximum flexibility.

A. Initialization and Setup

Only one input file and one output file are required by each of the Landsat Scheduler programs. These are initially opened by the main program and will remain open until processing of data has ceased.

The user is asked, by way of CRT prompts, to provide setup parameters to be used by the program (see Figure 2-1). The program will "echo" the setup parameters back to the CRT user, will write the first record on the output disk file using the mission start time, and will begin scheduling.

B. "Request Window" Scheduling

1. For the Landsat D odd-numbered orbits:

a. The program gets an LOS window using the DREAD sub-routine.

1) DREAD checks the PM flag (PMFLG) to see if a partial window was saved by a previous read in which a PM period subdivided an LOS window.

If so, the saved window is placed in WON and WOFF and the program continues.

If not, DREAD gets a new window by performing the following functions:

- Checks for alternating on-off status codes.
- Reads a window from the TDRS file until the window is between MSTART and MSTOP times, or until EOF is reached.

- Place window start and stop times in WON and WOFF ($WON \geq MSTART$ and $WOFF \leq MSTOP$).

- Sets the condition for the read flag (RFLG) to one of the following:

- 0 = "normal read" condition
- 1 = end of file encountered
- 2 = time read exceeded mission stop time
- 3 = window error (neither "on" nor "off" status)
- 4 = physical read error
- 5 = two consecutive end-of-LOS windows encountered
- 6 = two consecutive begin-of-LOS windows encountered

2) DREAD calls the PM schedule (PMSCHD) and PM generation (PMGEN) subroutines to check for a PM window and takes the appropriate action after setting the PM flag (PMFLG).

- PMSCHD checks to see if the current LOS window begins before the end of the current PM period (PMT2).

If not, a new PM window is generated and stored in a temporary storage array buffer (STACK). (NOTE: This STACK is the same buffer used to store "request window" results before sorting and writing them to the output disk file.) The program then continues to compare the new PM window with the current LOS window.

If the LOS window does begin before the end of the current PM period, then the start and stop times of both windows are compared.

- PMSCHD compares the current LOS window with the current PM window and adjusts, when necessary, the LOS window's WON and/or WOFF times to allow only the non-overlapping LOS window times to be available for scheduling. A PM flag (PMFLG) will be set to indicate the PM status condition:

- PMFLG = 0 Normal return - no PM occurring, or the overlapping LOS window was adjusted to the left or right of the PM window. The program will proceed.
- = 1 PM time was occurring for the entire LOS window, so the program will return to DREAD to get the next LOS window.
- = 3 PM time subdivided the LOS window, using the first portion for the current LOS window and saving the other portion for the next time when DREAD is called to get the next window. The program will proceed.

b. The LANDSA subroutine checks the read flag (RFLG) condition to assure that a "normal read" was made.

If not, the subroutine returns to the main program to exit, close files, and terminate.

If a "normal read" was made, then a "trial" TON and TOFF are established using the WON and WOFF returned from DREAD. This "trial" TON and TOFF are checked and adjusted, when necessary, to a final TON and TOFF (not less than the BECORB time or greater than the ENDORB time).

c. Next, LANDSA checks to see if the current "partial LOS window" (TON, TOFF) is wide enough (\leq DTIME) to schedule a R/T data relay.

If not, the start and stop times (T1, T2) will be set to a maximum negative value (-4000000.0). The program will proceed to store those scheduled results in the STACK storage buffer.

If the window is wide enough, the R/T data relay schedule times (T1, T2) will be computed as follows, using a random number between 0.0 and 1.0:

$$T1 = \text{Random No.} \times (\text{TOFF} - \text{TON} - \text{DTIME}) + \text{TON}$$

$$T2 = T1 + \text{DTIME}$$

d. The scheduled results for the Landsat D odd orbit (either the maximum negative values or the actual R/T data relay schedule times) will be stored in the temporary STACK storage buffer. The sign of the time values being placed in the buffer will be changed, negatively, to facilitate sorting the STACK later in descending order. At the same time, the following record-related data will be stored: Request ID, Request Type, Resource ID, and a complemented on-off status code. (NOTE: The on-off status codes being placed in the STACK are complemented so that when the STACK is written to the output disk file, the output file will represent a "resource availability" time line for the TDRS Scheduler to use. That is, when the Landsat Scheduler has scheduled a "request window," its beginning time record will be represented by an "off" status code and the ending time record will be represented by an "on" status code on the output disk file. The time remaining unscheduled (i.e., the time between an "on" and an "off" code) will be "resource availability" time to the TDRS Scheduler.)

2. For the Landsat D' odd-numbered orbits:*

a. LANDSA subroutine calls the DPRIME subroutine to schedule the start and stop times (DPT1, DPT2) for the Landsat D' odd orbit.

*Landsat Scheduler program LAND.D does not schedule a D' relay.

1) DPRIME immediately checks to see if the Landsat D odd orbit was scheduled.

If not, Landsat D' will not be scheduled. The start and stop times (DPT1,DPT2) will be set to the maximum negative value (-4000000.0). The program will proceed to store those scheduled results in the STACK storage buffer.

If the Landsat D odd orbit was scheduled, then Landsat D' will be scheduled based upon the Landsat D odd scheduled results. The schedule times for Landsat D' are computed as follows:

$$DPT1 = T1 + TOFFST$$

$$DPT2 = T2 + TOFFST$$

2) Once a D' relay has been scheduled, the start and stop times (DPT1,DPT2) are checked to see if either of them overlap or fall within a PM period.

If so, the D' odd orbit will not be scheduled. The start and stop times will be reset to the maximum negative value and the program will proceed to store those scheduled results in the STACK storage buffer.

If the D' start and stop times are outside the current PM time period, the program will proceed to store the scheduled results.

b. The scheduled results for the Landsat D' odd orbit (either the maximum negative values or the actual R/T data relay schedule times) will be stored in the STACK storage buffer as was done for the Landsat D scheduled results.

3. For the Landsat D and D'* even-numbered orbits:

a. The Landsat D even-numbered orbits will be scheduled and stored exactly the same as the odd-numbered orbits, except for the method of computing the even orbit start and stop times (T3,T4).

1) If the Landsat D odd orbit (T1,T2) was a schedulable event, then the even orbit (T3,T4) will be scheduled based upon the odd orbit as follows:

$$T3 = T1 + TORB$$

$$T4 = T3 + DTIME$$

Then, the start and stop times (T3,T4) will be checked against the current "partial LOS window" times (TON,TOFF) to insure that the scheduled times do not exceed the partial window boundaries. The start and/or stop times will be adjusted to the right or left when necessary.

2) If the Landsat D odd orbit (T1,T2) was not a schedulable event, the Landsat D even orbit (T3,T4) will be scheduled randomly, exactly as was done for the odd orbit.

$$T3 = \text{Random No.} \times (\text{TOFF} - \text{TON} - \text{DTIME}) + \text{TON}$$

$$T4 = T3 + \text{DTIME}$$

b. The Landsat D' even numbered orbit will be scheduled and stored exactly as was done for the Landsat D' odd orbit.*

NOTE: Figure 2-3 is a sample scheduling diagram depicting Landsat D odd and even orbit scheduling.

4. The records from the STACK storage buffer will be written to the output file and the program will proceed to process the next complete cycle of LANDSA subroutine until an error or end of file is encountered.

— After one complete cycle of the LANDSA subroutine, as described above, the first word of each of the STACK buffer records will contain the following:

<u>Rec. No.</u>	<u>Word 1 Contents</u>
1	A PM window start time
2	A PM window stop time
3	Landsat D odd orbit start time
4	Landsat D odd orbit stop time
5	Landsat D' odd orbit start time
6	Landsat D' odd orbit stop time
7	Landsat D even orbit start time
8	Landsat D even orbit stop time
9	Landsat D' even orbit start time
10	Landsat D' even orbit stop time

Thereafter, a maximum of two complete cycles (20 records) may be contained in the STACK buffer.

— At the end of each cycle through LANDSA, a flag (FLAG) will be set to indicate that a "normal read" condition (no errors or end of file was encountered, RFLG = 0) was maintained during the cycle.

When a "normal read" is no longer maintained, at any time during the cycle, LANDSA prepares to exit by setting the flag to clear the STACK of the remaining sorted records and writes them to the output disk file.

*Landsat Scheduler program LAND.D does not schedule a D' relay.

Figure 2-3. Landsat Scheduler sample scheduling diagram for Landsat D.

If a "normal read" condition is maintained, then the DWRITE sub-routine is called.

- DWRITE checks to see if ten or less records are stored in the STACK.

If so, the program holds those ten or less records in the STACK and returns to process the next complete "request window" scheduling cycle.

If more than ten records exist in the STACK, the records are sorted in descending order by chronological time. (Ten records will remain in the STACK to be chronologically sorted with the next cycle.) Those records exceeding the tenth one will be considered for writing to the output disk file, starting from the bottom of the STACK on up.

If the record was "schedulable," then the sign of the time word will be changed back to its original value. The record will be written to the output file.

If the record was "not schedulable," then the record will be skipped and the next one will be considered.

C. Closing and Summarization

LANDSA returns control to the main program when the final "request window"/scheduling results are written to the output disk file. The main program completes the processing by: (1) writing the last record on the output disk file using the mission stop time; and (2) completing run summary statistics and printing the summarization results to the CRT operator.

The files are closed (input and output) and the program returns control to the system monitor.

ORIGINAL PAGE IS
OF POOR QUALITY

2.4 SUBROUTINES

2.4.1 Common Variables - All Subroutines:

1. Subroutine Names:

The subroutines LANDSA, DREAD, DPRIME, DSTORE, DWRITE, PMSCHD, and PMGEN contain common variables. These common variables exist within a separate file, COMLST, and are inserted into the subroutines by means of '\$INSERT COMLST'.

2. Variables/Definitions:

DLTAT1 - Start of the potential scheduling period (in minutes).

DLTAT2 - Length of all potential scheduling periods (in minutes).

DPT1 - The current start time (in minutes) for a scheduled Landsat D' odd orbit relay.

DPT2 - The current stop time (in minutes) for a scheduled Landsat D' odd orbit relay.

DPT3 - The current start time (in minutes) for a scheduled Landsat D' even orbit relay.

DPT4 - The current stop time (in minutes) for a scheduled Landsat D' even orbit relay.

DTIME - The duration of a real/time data relay (in minutes).

FLAG - A write status flag:

NOCLR = 0, a normal clear call to write a portion of the STACK to the output disk file.

ICLR = 1, a final clear call to empty the entire STACK.

INAME - The input file name.

IOFF - Contains the numeric value of 2 which represents "off".

ION - Contains the numeric value of 1 which represents "on".

IR - The current record number counter for the STACK.

ISTFLG - A store flag to indicate what kind of data is being stored. If the value is:

0 = a request window.

1 = a PM window.

KEY1 - Represents a numeric value of 1.

KEY2 - Represents a numeric value of 2.

KEY3 - Represents a numeric value of 3.

KEY4 - Represents a numeric value of 4.

LINE - The current line counter for the output data printed to the CRT operator.

MAX - Represents the maximum value of -4000000.0.

MSTART - The mission start time (in minutes).

MSTOP - The mission stop time (in minutes).

NRFIL - The cumulative count for the number of requests filled.

NRUNF - The cumulative count for the number of requests unfilled.

ONAME - The output file name.

PMDUR - Duration time of each PM period (in minutes).

PMFLG - A PM status flag:
 0 = normal return
 1 = PM occurring
 3 = PM subdivided an LOS window

PMFREQ - Time between successive PM periods (in minutes).

PMTIME - The cumulative total PM time (in minutes).

PMT1 - The start time of a Preventative Maintenance (PM) window.

PMT2 - The stop time of a Preventative Maintenance (PM) window.

PMSTRT - The start time of the first PM period (in minutes).

RBUF - The 32-word buffer for the current I/O records.

RFLG - Represents a read flag status:
 0 = Normal read.
 1 = EOF.
 2 = Time read exceeded mission stop time.
 3 = Window error (not "on" or "off").
 4 = Physical read error.
 5 = Two consecutive end-of-LOS windows.
 6 = Two consecutive begin-of-LOS windows.

RESID - ID code to represent the type of resource (i.e., TDRS-E, TDRS-W).

REQID - ID code to represent the request (i.e., Landsat D, Landsat D', PM).

REQTYP - ID code to represent the type of request (i.e., Real/Time relay, T/R dump).

RQID - The current request ID being scheduled.

STACK - The temporary storage buffer for request windows before writing them to the output disk file.

SWOFF - A saved stop time created when PM subdivided an "LOS window".

SWON - A saved start time created when PM subdivided an "LOS window".

TOFFST - The offset time period between Landsat D and D' (in minutes).

TORB - The length of the orbital period (in minutes).

WOFF - The stop time for the current LOS window.

WON - The start time for the current LOS window.

2.4.2 Subroutine LANDSA

1. Description:

LANDSA schedules an odd and even orbit R/T relay for Landsat D and D'. If an odd or even orbit cannot be scheduled on Landsat D, a maximum value (-4000000.0) will be stored in all time words to prevent those events from going to the output file.

Because some Landsat-TDRS contacts last for more than one Landsat orbit, provisions are made to limit the line-of-sight window to a single orbital period by calculating a beginning orbit time (BEGORB) and ending orbit time (ENDORB). Any remainder of a window beyond the ENDORB time will be considered for the next orbit before input is read again.

The variables WON and WOFF are used to hold the on-off window pair as read from the input disk file. The window for an orbit currently being scheduled will be held in the variables TON and TOFF ($TON \geq BEGORB$ and $TOFF \leq ENDORB$).

After each Landsat D schedule (odd or even orbit), the scheduled results will be printed to the CRT operator, and LANDSA will call the subroutine DPRIME to schedule the Landsat D' orbits.*

2. Variables/Definitions:

a. Arguments:

-

b. Locals:

BEGORB - Beginning orbit time (in minutes) for the current orbit.

*Because the Landsat Scheduler program LAND.D does not schedule a D' relay, LANDSA will not call the DPRIME subroutine for scheduling.

DLT3 - Difference (in minutes) between an even orbit start time and its adjusted start time.

DLT4 - Difference (in minutes) between an even orbit stop time and its adjusted stop time.

ENDORB - Ending orbit time (in minutes) for the current orbit.

ORBNO - Number of the current orbit (1,2,...N).

R - Current random number between 0.0 and 1.0.

ST3 - A saved even orbit start time (in minutes).

ST4 - A saved even orbit stop time (in minutes).

TOFI - The current "partial LOS window" stop time (in minutes).

TON - The current "partial LOS window" start time (in minutes).

T1 - The current start time (in minutes) for a scheduled Landsat D odd orbit relay.

T2 - The current stop time (in minutes) for a scheduled Landsat D odd orbit relay.

T3 - The current start time (in minutes) for a scheduled Landsat D even orbit relay.

T4 - The current stop time (in minutes) for a scheduled Landsat D even orbit relay.

3. Input Variable Names:

DLTAT1, DLTAT2, DTIME, ICLR, LINE, MAX, MSTART, NOCLR, REQID(1), REQID(2), RFIG, TORB, WOFF, WON.

4. Output Variable Names:

DPT1, DPT2, DPT3, DPT4, FLAG, LINE, NRFIL, NRUNF, REQID(2), REQID(3), T1, T2, T3, T4.

5. Called by:

Main Program.

6. Calls:

Subroutine DREAD, DSTORE, DPRIME, DWRITE.

2.4.3 Subroutine DPRIME (TB,TE)

1. Description:

DPRIME schedules an odd or even orbit relay for Landsat D', based upon the Landsat D scheduled results.

- If Landsat D was not scheduled (-4000000.0 in the Landsat D time words), then Landsat D' will not be scheduled. (Storing -4000000.0 in all the time words will prevent those events from going to the output file).

- If Landsat D was scheduled, then D' will be scheduled, and the start and stop times will be checked for exceeding the mission stop time or for occurring during a PM time period. If either of these are true, then the maximum negative value (-4000000.0) is returned as the D' scheduled results.

DPRIME prints the Landsat D' scheduled results to the CRT operator, along with the appropriate program message if the event was not scheduled.

2. Variables/Definitions:

a. Arguments:

TB - Coming into the subroutine, TB is the start time of the Landsat D odd or even orbit. Going out of the subroutine, TB is the start time of the Landsat D' odd or even orbit.

TE - Coming into the subroutine, TE is the stop time of the Landsat D odd or even orbit. Going out of the subroutine, TE is the stop time of the Landsat D' odd or even orbit.

b. Locals:

3. Input Variable Names:

LINE, MAX, MSTOP, NRFIL, NRUNF, PMT1, PMT2, TB, TE,
TOFFST

4. Output Variable Names:

LINE, NRFIL, NRUNF, TB, TE

5. Called by:

Subroutine LANDSA.

6. Calls:

2.4.4 Subroutine DREAD

1. Description:

DREAD performs the following:

— Checks the condition of the PM flag (PMFLG) and takes the appropriate action:

PMFLG = 0, Normal return — no PM occurring, or the overlapping LCS window was adjusted to the left or right of the PM window. The program will proceed.

= 1, PM time was occurring for the entire LOS window, so the program will return to DREAD to get the next LOS window.

= 3, PM time subdivided the LOS window, using the first portion for the current LOS window and saving the other portion for the next time when DREAD is called to get the next window. The program will proceed.

— Reads the input file to get next window until it is between MSTART and MSTOP or until error or EOF is encountered. Window start and stop times are placed into WON, WOFF, ($WON \geq MSTART$ and $WOFF \leq MSTOP$).

— Makes checks on the window record and, according to the checks made, sets a read flag (RFLG) condition as follows:

RFLG = 0, "normal read" condition.

= 1, EOF was encountered.

= 2, time read exceeded mission stop time.

= 3, window error (status neither "on" nor "off").

= 4, physical read error.

= 5, two consecutive end-of-LOS windows encountered.

= 6, two consecutive begin-of-LOS windows encountered.

2. Variables/Definitions:

a. Arguments:

b. Locals:

IFLG - Flag to represent the portion of the window read. Initially it is preset to -1 to indicate at mission start time. After the first record is read, IFLG will be set as follows:

0 = an end-of-LOS window record was read.

1 = a begin-of-LOS window record was read.

TIME - Total time of an event in minutes.

3. Input Variable Names:

IBUF, IOFF, ION, MSTART, MSTOP, PMFLG, RBUF, SWOFF, SWON, WOFF, WON.

4. Output Variable Names:

PMFLG, RFLG, WOFF, WON.

5. Called by:

Subroutine LANDSA

6. Calls:

Subroutine PMSCHD

2.4.5 Subroutine PMSCHD (TB.TE)

1. Description:

PMSCHD checks the current LOS window against the current PM window whose stop time is greater than the LOS window's start time. It adjusts the LOS window's start and/or stop times, when necessary, to avoid any overlapping times and sets a PM flag (**PMFLG**) condition as follows:

PMFLG = 0, No overlapping exists between PM and LOS windows.

= 1, LOS window start and stop times completely overlapped with PM times, so will return to get a new window.

= 3, PM time subdivided an LOS window, using the first portion for the current LOS window and saving the other portion for the next time when **DREAD** is called to get the next window.

2. Variables/Definitions:

a. Arguments:

TB -- start time of the current LOS window (WON).

TE -- stop time of the current LOS window (WOFF).

b. Locals:

--

3. Input Variable Names:

PMT1, PMT2, TB, TE

4. Output Variable Names:

ISTFLG, PMFLG, PMT1, PMT2, TB, TE, SWOFF, SWON

5. Called by:

Subroutine DREAD

6. Calls:

Subroutines PMGEN, DSTORE.

2.4.6 Subroutine PMGEN

1. Description:

PMGEN generates a "preventative maintenance window" (i.e., a start and stop time when preventative maintenance for computer hardware will be occurring).

2. Variables/Definitions:

a. Arguments:

--

b. Locals:

--

3. Input Variable Names:

MSTART, MSTOP, PMDUR, PMFREQ, PMSTRT

4. Output Variable Names:

PMTIME, PMT1, PMT2

5. Called by:

Subroutine PMSCHD

6. Calls:

-

2.4.7 Subroutine DSTORE (TB,TE)

1. Description:

According to the status of the store flag (ISTFLG), DSTORE places the appropriate window start and stop times in a storage buffer (STACK) along with additional unique identification for each record.

ISTFLG = 0, a scheduled relay window is being stored.

= 1, a PM window is being stored.

In both of the cases, above, the signs are changed on the start and stop times to effect the desired descending order sort in subroutine DWRITE.

2. Variables/Definitions:

a. Arguments:

TB -- start time for the current odd or even Landsat D, D', or PM scheduled results.

TE -- stop time for the current odd or even Landsat D, D', or PM scheduled results.

b. Locals:

-

3. Input Variable Names:

IOFF, ION, IR, ISTFLG, TB, TE, RBUF(2), REQID(3), REQTP, RESID, RQID

4. Output Variable Names:

IR, STACK(1,IR), STACK(2,IR), STACK(3,IR), STACK(4,IR), STACK(5,IR), STACK(6,IR)

5. Called by:

Subroutine LANDSA, PMSCHD

6. Calls:

--

2.4.8 Subroutine DWRITE

1. Description:

DWRITE writes the schedulable events to the output disk file.

-- Tests the FLAG status:

FLAG = 0, normal call -- only empty the STACK down to the tenth record.

= 1, clear the entire STACK.

-- Tests the number of records in the STACK. If less than or equal to ten records in the STACK, return. If greater than ten records, calls CRSRTR subroutine to sort the STACK in descending order.

-- Beginning with the bottom of the STACK, checks the records' time words for being greater than zero. If the times are greater than zero, the records are skipped. If not, the time words are converted to seconds and milliseconds. Their signs are changed back to their original ones and the records are written to the output disk file.

2. Variables/Definitions:

a. Arguments:

--

b. Locals:

IPMSEC -- Total PM seconds (integer).

KEYS -- The array of indices for the order in which words are to be sorted.

NCREC -- Number of records to be sorted.

NKEYS -- Number of keys to be sorted.

NWD -- Number of words to be sorted.

PMMSEC -- Total PM milliseconds (real).

PMSEC - Total PM seconds (real).

3. Input Variable Names:

**FLAG, IOFF, ION, IR, NOCLR, RBUF(32), REQID(3),
(STACK(J,IR), J = 1,6)**

4. Output Variable Names:

IBUF(32), IR, KEYS, NCREC, NKEYS, NWD, RBUF(32)

5. Called by:

Subroutine LANDSA

6. Calls:

Subroutine CRSRTR.

3. TDRS SCHEDULER

3.1 GENERAL

This Scheduler primarily schedules tape-recorder dumps from satellite payloads through a system of TDRS's. It uses, as input, the output time line files from the Landsat Scheduler program to determine the available time intervals remaining for scheduling data relays after the Landsat real-time relays and preventive maintenance. DSDS is used to generate LOS time lines for the satellites that record data on tape for subsequent T/R dumps, as well as for real-time data relays. See Table 3-1 for LOS file formats.

Each dump can be made at any time within a fixed window which will be defined for each satellite. The TDRS relay channels are available resource windows during the time frames when the TDRS is not busy. The windows or time periods during which a dump must occur are referred to as request windows. The line of sight time lines for the satellites are the periods of time within which requests can be filled.

The program will write an output disk file with each record containing a message for each significant status change. Refer to Appendix B.3 for an example of a test run output file. The three message types are described as follows:

- * (1) "Request scheduled" messages will include:
 - a. time period (begin time and end time)
 - b. request ID
 - c. resource ID
 - d. current priority rating
 - e. resource type code
 - f. request type code
 - g. message type code = 1
- (2) "Unscheduled request" messages will include:
 - a. time period
 - b. request ID
 - c. current priority rating
 - d. request type code
 - e. message type code = 2
- (3) "Unused resource" messages will include:
 - a. time period
 - b. resource ID
 - c. resource type code
 - d. message type code = 3

AREAEXPAND will be used to generate various plots for detailed analysis and reports (see Appendix B.5).

*NOTE: To be compatible with the plot software, the output message file must contain a record for each status change.

TABLE 3-1. TIME LINE FORMATS (32 WORDS)

Req./Res. LOS Time Line File (DSDS)	Scheduled Time Line File (Landsat)	Scheduled Time Line File (TDRS Scheduler)
1. Time - Sec. INTEGER 2. Time - Mil. REAL 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. ON/OFF Code - INTEGER 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32.	1. Time - Sec. INTEGER 2. Time - Mil. REAL 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. Req. ID - INTEGER 15. Req. Type - INTEGER 16. ON/OFF Code - INTEGER 17. Res. ID - INTEGER 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32.	1. Time - INTEGER 2. Time - Mil. REAL 3. 4. 5. 6. 7. 8. 9. 10. Msg. Type - INTEGER 11. Res. Type - INTEGER 12. Priority - INTEGER 13. ON/OFF Cmpl. - INTEGER 14. Req. ID - INTEGER 15. Req. Type - INTEGER 16. ON/OFF Code - INTEGER 17. Res. ID - INTEGER 18. 19. Sum1 - REAL 20. Sum2 - REAL 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32.

*NOTE - Sum1 and Sum2 depend on message type code.

For code=1, Sum1= Current time scheduled for this requestor.
Sum2= Current time requested by this requestor.

For code=2, Sum1= Current request time not filled for requestor.
Sum2= Current time requested by this requestor.

For code=3, Sum1= Current time wasted for this resource.
Sum2= Current resource time available for this resource.

3.2 INTERACTIVE CAPABILITIES

To execute the TDRS Scheduler, the operator must first manually generate setup files, using the FILESET program (see Appendix C) or the Editor, to specify and parameterize the resources and requestor requirements. See Table 3-2 for a description of the data structure for the setup files, and Appendix B.1 for examples of resource and requestor setup files. Note that the filenames in record 1 of the resource setup file are the Landsat scheduled time line files generated as output by the Landsat Scheduler, whereas the LOS time line filenames in record 4 are LOS time lines generated by DSDS for each requestor/resource combination. Refer to Table 3-1 for LOS time line file formats.

During program initialization operator input parameters include mission start, mission stop and statistical stop times, resource setup filename/s, requestor setup filenames/s, binary output filename and detailed print output filename. Refer to Appendix B.4 for error messages which may appear when appropriate during a test run along with subsequent program actions. The initialization and run time procedures for the TDRS Scheduler follow:

- (1) Sign-in on the PRIME 400 computer using the appropriate user file directory (UFD), where the files to be accessed are located. Refer to Appendix B.2 for an example of an execution run.
- (2) For a hard copy of everything that appears on the CRT screen during a test, enter 'COMO filename' for eventual spooling to the line printer.
- (3) To initiate program execution, enter 'SEG #TDRS.SCHED'.
- (4) Respond to the prompts for time parameters with real numbers (using a decimal point) in minutes:

Mission start, mission stop and statistical stop.
Statistical time interval.
- (5) Respond to the prompts for filenames with a maximum of 6 alphanumeric characters, starting with an alphabetic:

Output filename.
Detailed output print filename.

Respond to the prompt for initial interrupt/restart time with a real number.
- (6) Respond to the prompts for filenames with a maximum of 6 alphanumeric characters, starting with an alphabetic:

Resource setup filename.
Requestor setup filename.

Respond to the prompts for another resource and/or requestor setup filename. If none, enter carriage return for each prompt to continue.

TABLE 3-2. TDRS SCHEDULER RESOURCE AND REQUESTOR SETUP FILES

Description	Format	Type	RAVAIL Res. Array	RQTBL Req. Array	Common Variable
Record 1 Unique ID Active(0)/Inactive (#) Code 1 Filename (6 Char) File Description	I6 I6 2A4 5A4	INTEGER INTEGER ALPHA ALPHA	RESID(N1) RESAI(N1) RESFN(2,N1) RESDS(5,N1)	REQID(N2) REQAI(N2) REQFN(2,N2) REQDS(5,N2)	ID IFILE(2)
Record 2 Unique ID Priority Code (1,2) Definition Param.	I6 I6 5F12.3	INTEGER INTEGER REAL	RESID(N1) RESPR(N1) RESPAR(M1,N1)	REQID(N2) REQPR(N2) REQPAR(M2,N2)	ID
Record 3 - If req'd. Unique ID 3 Variable Length Relay Param. (4)	I6 4F12.3	INTEGER REAL		REQID(N2) REQPAR(M2,N2)	ID RMEAN, RMIN. RMAX, SIGMA
Record 4 Unique ID 4 Req./Res. LOS Time Line Filenames (6 Char)	I6 10(2A4)	INTEGER ALPHA		REQID(N2) RQSFN(2*N1,N2)	ID IFILE(2)
Record 5 - If Req'd. Unique ID 5 No. Dependent ID's Dependent ID List (1 to 4) Dependent ID Time Delays (Deg.)	I6 I6 4F12.3 4F12.3	INTEGER INTEGER REAL REAL		REQID(N2) REQDID(1,N2) REQDID(M3+1,N2) DELAY(M3)	ID NIDREQ(N2) DIDREQ(M3+1,N2)

TABLE 3-2. (Concluded)

- N1 = No. of resources in RAVAIL
 N2 = No. of requestors in RQTBL
 M1 = No. of parameters to describe this resource type
 M2 = No. of parameters to describe this requestor type
 M3 = No. of parameters to describe this resource type required.
1. Scheduled time line filename from LANDSAT Scheduler. For iterative test run, TDRS Scheduler output filename.
 2. By definition:
 REQPAR (1,N2) = request type code (1 = R/T data relay, 2 = T/R dump).
 REQPAR (2,N2) = p, orbital period.
 REQPAR (3,N2) = n, one request window generated per n orbits (request types 3 or 5).
 = m, one request window generated per m minutes (request types 4 or 6).
 REQPAR (4,N2) = k1, window parameter.
 REQPAR (5,N2) = k2, window parameter.
 REQPAR (6,N2) = DTIME, request duration.
 Refer to Appendix B.6 for a detailed description of request algorithm generation.
 3. If request type code >100, four variable length relay parameters (all requestor types) as follows:
 REQPAR (7,N2) = Mean value.
 REQPAR (8,N2) = Minimum value.
 REQPAR (9,N2) = Maximum value.
 REQPAR (10,N2) = Standard deviation.
 4. LOS file for each requestor with every resource available combination.
 5. Dependent IDs for requestor types 4 or 6 only. ID list must follow parent ID in setup file.

(7) Following the printed message "Scheduling in Progress", the Scheduler will attempt to schedule data relays. See Section 3.3, Program Flow, for statistics for each requestor to be printed on the CRT for each statistical time interval. When either all resources or all requestors become inactive or when statistical stop time is reached, all records will be written to the output file, and total statistics for each requestor will be printed to the CRT screen.

(8) Another more complete set of statistics for requestors and resources is computed and displayed on the CRT. Refer to Section 3.3, Program Flow, for a description of the statistics.

(9) If a 'COMO' file was made, enter 'COMO -END'. Then enter 'SPOOL filename' to get a hard copy of the file on the line printer.

(10) To convert the binary output file to a readable ASCII format, enter 'SEG #NEW OUTPUT'. Respond to the prompts for output filename, and either a spooling filename or carriage return for an ASCII printout to the CRT.

(11) To make an iterative run with the remaining available resource/s time intervals, manually generate another resource setup file with FILESET or with the Editor. Add 2000 to the resource ID, taken from the previous output file, to indicate to the Scheduler that this will be an iterative run. Note that the remaining available resources may be used in conjunction with other resources.

Manually generate a new requestor setup file/s for other requestor requirements. Then begin program execution with step 2).

(12) If desired, the operator may run AREAEXPAND to generate various plots for detailed analysis and reports.

3.3 PROGRAM FLOW

The Scheduler block diagram indicates a general program flow (see Figure 3-1):

(1) Manually generated setup files are read into control arrays which specify and parameterize the resources and requestor requirements. Resources and requestors require a minimum of two separate files; data for a given resource will be input by 2 disk records, whereas data for each requestor will be input by 3 to 5 disk records as required. The two pseudo control arrays, RAVAIL, which describes the resources and RQTBL, which describes the requestor requirements, are used to control the scheduling sequence. Each pseudo array is comprised of matrices which parameterize each resource or each requestor. See Table 3-2 for control array variable names and composition, and Section 3.4, Subroutines, for parameter and variable definitions.

(2) The first resource window available for scheduling is read and stored into RAVAIL, by row, for each resource (subroutine RESRD). RESRD calls subroutine FILRD, which reads a computed number of resource available windows into the DIN array when necessary according to a file position pointer, and returns the first window to the calling routine. Subsequent calls return the next available resource window until it becomes necessary to read the next group of records from that file until end of file.

(3) A subroutine called REQGEN is used to generate various types of request requirements based on the request type code and request description parameters which may designate a variable request time. The type code is used to select the appropriate algorithm to generate requests based on the accompanying parameters. See Appendix B.6 for a description of the request generation algorithms. REQGEN places the generated request requirements into RQTBL in the appropriate row.

(4) The resource available window with the earliest start time is selected for 'next' resource to be scheduled (subroutine EARWND). At this time (start of the window) there may be requests which cannot be scheduled. They are replaced by the next requirement from the same requestor (subroutine REQUPD). For statistical computations, the summation of request time periods lost is stored, and unfilled request messages are generated and stored in the output disk file (subroutine MSG2).

(5) If the interrupt/restart time has been reached, the resource and requestor control arrays are updated and the files are positioned properly (see POSREQ and POSRES in Section 3.4, Subroutines). Also the next interrupt/restart time is loaded via operator input.

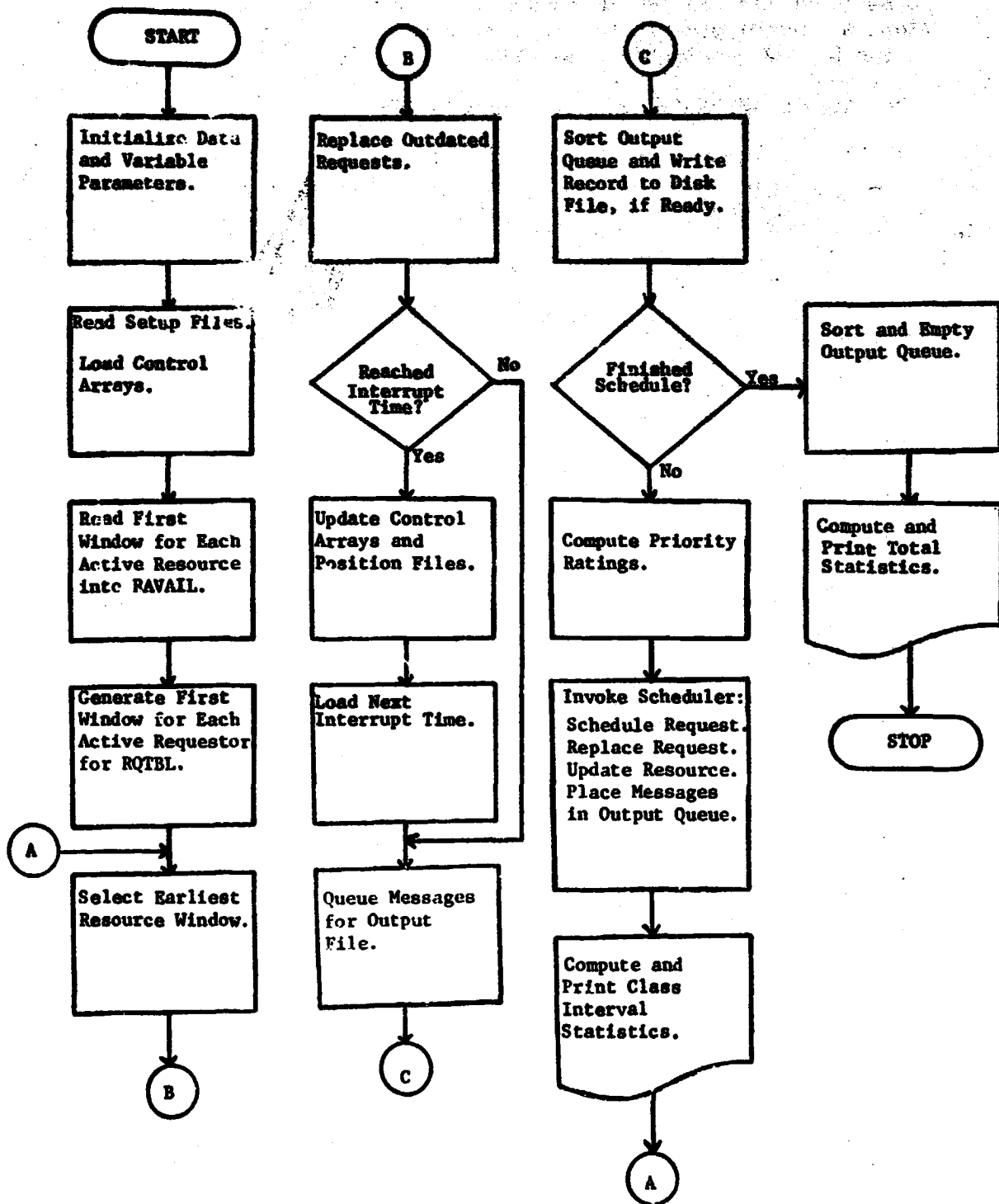


Figure 3-1. TDRS Scheduler functional block diagram.

(6) If there are any resources left to be scheduled and requests to be filled and the resource window is between mission start and mission stop, a current priority rating is computed and stored for each request in the RQTBL (subroutine PRIORT).

Two levels of priority are considered:

Level 1 - all active requests with window reference time less than the start time of the current earliest resource available window; i.e., $T_R < W_{on}$.

Level 2 - all other requests.

(7) To avoid wasting resources, the Scheduler (subroutine SCHEDL) first attempts to schedule a request at the resource window opening by considering requests in priority order and in RQTBL order (subroutine OPTSCH). If all conditions are met in OPTSCH, subroutine CKLOS checks line of sight between requestor and resource. If the request is scheduled, a scheduled request message is sent and RQTBL is updated (subroutine MSG1), and the resource in RAVAIL is updated (subroutine UPDRES).

If the optimum schedule cannot be made and resources must be wasted (subroutine MSG3), the waste is minimized by scheduling the earliest request available that fits (subroutine CKLOS2) by priority order using subroutine EARSCH. If a request is then scheduled, MSG1 and UPDRES are called.

If the entire resource cannot be scheduled, an unused resource message is sent (subroutine MSG3), and RAVAIL is updated (subroutine UPDRES).

Statistics are computed (subroutine STATSV) and printed to the operator (subroutine CLSSPR) for each requestor throughout the scheduling sequence. Statistics printed on the CRT according to the statistical class interval include:

- Number of unfilled dumps
- Number of scheduled dumps
- Time lost in minutes
- Time scheduled in minutes.

(8) When all the resource files or the requests are depleted (all resources or all requestors inactive) or statistical stop time is reached, all status change messages are written to the output file from the output queue (subroutine STOREQ) in chronological order (subroutine CLEARQ which calls WRITEQ). The following statistics, again for each requestor, are printed to the screen (subroutine TOTLPR):

Total unfilled dumps

Total scheduled dumps

Total time lost

Total time scheduled.

Other statistics, for the combined requestors and the combined resources, are printed to the operator (in minutes):

Total requested time

Total unfilled request time

Total available resource time

Total scheduled time

Total wasted resource time

Percentage of total requests filled

Percentage of total resource utilization

Number of scheduled requests

Number of unfilled requests

Number of unused resources.

3.4 SUBROUTINES

3.4.1 Common Variable Definitions

RAVAIL Resource Control Array

RESID(N1) = ID.

RESAI(N1) = Active (0)/inactive (≠0) code.

RESFN(2,N1) = Time line filename.

RESDS(5,N1) = File description.

RESPR(N1) = Priority (not used).

RESPAR(M1,N1) = Type code and parameters (not used).

RESW(2,N1) = Time interval of available resource from TON to TOFF.

RESPTR(N1) = Block pointer.

RESSUB(N1) = Sub block pointer.

RESPOS(N1) = File read position pointer.

where N1 = No. of resources.

M1 = No. of description parameters.

RQTBL Requestor Control Array

REQID(N2) = ID.

REQAI(N2) = Active/inactive code.

REQFN(2,N2) = Time line filename (not used).

REQDS(5,N2) = File description.

REQPR(N2) = Priority.

REQPAR(M2,N2) = Type code and parameters.

RQRSFN(2*N1,N2) = Requestor/resource LOS time line filenames.

REQDID(M3+1,N2) = No. of dependent ID's for this parent and ID list.

DELAY(M3) = Delays between dependents and parent ID (Degrees).

REQW(2,N2) = Time interval during which resources are needed.

REQPTR(N2) = Block pointer into DIN array.

REQSUB(N2) = Sub block pointer into DIN array.

TD(N2) = Used to compute TR. If last request was not filled, TD = end of request window. Otherwise, TD = start of last request filled for a requestor.

TR(N2) = Computed request window reference time,
(TR+k1, TR+k2) where k1,k2 = window parameters;
i.e., offsets from the reference time.

See Appendix B.6 for requestor algorithm descriptions for each requestor type.

R(N2) = Pseudo random number between 0.0 and 1.0.

LOSPTR(N1,N2) = Block pointer into DIN array.

LOSSUB(N1,N2) = Sub block pointer into DIN array.

RQRSPO(N1,N2) = File read position pointer.

where N2 = No. of requestors.

M2 = No. of description parameters.

M3 = No. of dependent IDs.

OTHERS

BINT,EINT = Schedulable interval computed in CKLOS2 subroutine. Intersection of resource window, requestor window, and requestor-resource LOS time line.

BLKPTR = Block pointer for last record read from a given disk file (DIN array pointer).

BUFFER(10) = Array for individual status messages before they are transferred to the output queue.

CLASS = Statistical class time interval.

CREQ(N2) = Array containing all requests that could be scheduled within the current window (WON, WOFF) - subr. EARSCH.

DATFIL(2) = Detailed data print filename.

DIN(10000) = Array into which LOS time line records are read, according to file position format. Each block as follows:

WORD 1 = Window ON time.

WORD 2 = Window OFF time.

WORD 3 = Resource ID. If applicable.

WORD 4 = Requestor ID. If applicable.

WORD 5 = Request type. If applicable.

DTIME = Duration of a request, within a request window -- also REQPAR (6,1).

ID = Resource or requestor ID.

IFILE = Line of sight time line filename.

IFILE = Line of eight time line filename.

IK = Dynamic pointer for DIN array into which line of eight time line file records are read.

IMSG1 = Count of scheduled request messages.

IMSG2 = Count of unfilled request messages.

IMSG3 = Count of unused resource messages.

INIT = Flag to update statistical class interval.

INTFL = Interrupt/restart flag.

IRQID = Requestor ID.

IRQTP = Requestor type.

IRSID = Resource ID.

I1 = Requestor row number in RQTBL.

I2 = Resource row number in RAVAIL.

L(50) Used by random number generator function DRAND to provide 50 distinct pseudo random number streams.

MSTART = Mission start time (in minutes).

MSTOP = Mission stop time (in minutes).

MSTP = Statistical stop time for completion of statistical computations before actual MSTOP (in minutes). Generally MSTOP should exceed MSTP by an amount \geq the longest scheduling period.

NCDT(N2) = Total number of scheduled dumps for each requestor.

NOREC = Computed number of records allocated in the DIN array per time line file access.

NPLEVS = Number of priority index table levels.

NPOS = Time line file position pointer used in FILRD subr.

NQ = Number of messages in output queue.

NRQ = Number of requestors.

NUNFT(N2) = Total number of unfilled dumps for each requestor.

OPFILE(2) = Output filename.

PRATE = Priority level/rating currently used for scheduling, (0 = highest, 1 = next).

QTIME = Latest time to be written from output queue. Normally set to the earliest request window closing time; set to PAB in CLEARQ.

QUEUE(32, 150) = Array where output records are temporarily stored to insure output file is written in chronological order.

REQUNF(N2) = Dump minutes lost for each requestor.

RFLG = Disk file read flag. (0 = successful read, 1 = EOF) - subr. FILRD.

RNCD(N2) = Number of scheduled dumps for each requestor within a class interval.

RNUNF(N2) = Number of unfilled dumps for each requestor within a class interval.

ROWK = Row number of earliest active resource available window in RAVAIL.

RQNFT(N2) = Total unfilled dump minutes for each requestor.

SIZ = Number of words for each DIN array block.

STATUS = Has 0 value if scheduling subroutine unsuccessful. If a request is scheduled, Status contains RQTBL row number of scheduled request.

SUBPTR = Sub block pointer for last record read from a given disk file - DIN array pointer.

TAVL = Total available time in minutes for all resources combined.

TB,TE = Beginning and end times for unused resource.

TBEGIN,TEND = Beginning and end times for unfilled request.

TI = Operator input interrupt/restart time.

TIMSCD(N2) = Scheduled dump minutes for each requestor.

TIMSDT(N2) = Total scheduled dump minutes for each requestor.

TIM1, TIM2 = Statistical time interval begin and end times.

TMIN = Software end-of-file indicator.

TON, TOFF = Line of sight time line window.

TREQ = Total requested time in minutes.

TSCH = Total scheduled time in minutes.

TUNF = Total unfilled request time in minutes.

TWST = Total wasted resource time in minutes.

T0, TN = Beginning and end ($T0 + DTIME$) times for scheduled request.

T1, T2 = Request data relay window for optimum schedule attempt, beginning at start of resource available window.

WON, WOFF = On and off times for current earliest resource window/ i.e., the begin and end times for the available resource with next available time for scheduling.

3.4.2 Subroutine CKLOS (I1,I2,T1,T2,CHECK)

1. Description:

CKLOS checks line of sight between requestor and resource for optimum relay between times T1 and T2. If the requestor I1 can see the resource I2 between T1 and T2 continuously, CHECK flag is set to 1; otherwise to 0. If end of file is reached ($RFLG = 0$ from FILRD) for requestor-resource time line file from DSDS, requestor becomes inactive in RQTBL.

2. Input Variable Names:

I1, I2, TON, TOFF, T1, T2, RFLG, RQRSFN(2,I1),
BLKPTR, SUBPTR, NPOS.

3. Output Variable Names:

CHECK, LOSPTR(I2,I1), LOSSUB(I2,I1), RQRSPO(I2,I1)
REQAI(I1).

4. Called by:

OPTSCH.

5. Calls:

FILRD.

3.4.3 Subroutine CKLOS2(I1,I2,BINT,EINT,CHECK)

1. Description:

CKLOS2 determines if the requestor/resource LOS simultaneously intersects the requested time interval and the earliest resource available window. If the requestor I1 can see the resource I2 between BINT and EINT continuously, CHECK flag is 1; otherwise 0. If end of file is reached (RFLG = 0 from FILRD) for requestor-resource time line file from DSDS, requestor becomes inactive in RQTBL.

2. Input Variable Names:

I1, I2, TON, TOFF, RFLG, RQRSFN(2,I1), BLKPTR, SUBPTR, NPOS.

3. Output Variable Names:

CHECK, LOSPTR(I2,I1), LOSSUB(I2,I1), RQRSPO(I2,I1), REQAI(I1), BINT, EINT.

4. Called by:

EARSCH.

5. Calls:

FILRD.

3.4.4 Subroutine CLEARQ

1. Description:

CLEARQ calls WRITEQ to write all records in the output queue to the output disk file.

2. Input Variable Names:

None.

3. Output Variable Names:

QTIME = PAB (plus all bits).

4. Called by:

MAIN.

5. Calls:

WRITEQ.

3.4.5 Subroutine CLMSG3 (ROWK)

1. Description:

CLMSG3 sends an unused resource message for ROWK in SVMSG3 array to the output queue if it contains a valid window. SVMSG3 saves unused resource messages which may be contiguous with later messages.

2. Input Variable Names:

SVMSG3(2, ROWK), ROWK.

3. Output Variable Names.

None.

4. Called by:

SCHEDL.

5. Calls:

MSG3.

3.4.6 Subroutine CLSSPR

1. Description:

CLSSPR computes and prints the following statistics for each requestor for each statistical class interval that begins prior to the statistical stop time: number of unfilled dumps, number of scheduled dumps, dump time lost in minutes, dump time scheduled in minutes.

2. Input Variable Names:

NRQ, TIM1, TIM2, MSTP, NCD(NRQ), NUNF(NRQ), NUNFT(NRQ), RQUNFT(NRQ), REQUNF(NRQ), NCDT(NRQ), TIMSDT(NRQ), TIMSCD(NRQ).

3. Output Variable Names:

RNCD(NRQ), RNUNF(NRQ), NUNFT(NRQ), RQUNFT(NRQ), NCDT(NRQ), TIMSDT(NRQ).

4. Called by:

CLEARQ, STATSV.

5. Calls:

EXIT.

3.4.7 Subroutine CRSRTR(IR,KEYS,NKEYS,NWD,NREC)

1. Description:

CRSRTR sorts a real array in core according to arguments passed to it.

2. Input Variable Names:

IR = array name,
NREC = number of records to be sorted.
KEYS = array of indices for order in which words are sorted,
NKEYS = number of KEYS,
NWD = number of words to be sorted.

3. Output Variable Names:

IR.

4. Called by:

WRITEQ, EARSCH.

5. Calls:

None.

3.4.8 Subroutine EARSCH(STATUS,ROWK,N2,BINT)

1. Description:

EARSCH attempts to schedule the earliest possible active request from RQTBL according to priority after optimum scheduling was not possible. All eligible requests go into CREQ array. If the earliest possible schedule is made, STATUS = request row number in RQTBL. If request cannot be scheduled for this resource, STATUS = 0. See Appendix B.6 for request generation algorithms.

2. Input Variable Names:

REQAI(N2), N2, REQPR(N2), WON, WOFF, REQPAR(6,N2),
REQW(1,N2), CHECK, BINT, EINT, NPLEVS, ROWK.

3. Output Variable Names:

I = request row number in RQTBL, ROWK, BINT, EINT,
CREQ(N2), NREC, KEYS, NKEYS, NWD, STATUS.

4. Called by:

SCHEDL.

5. Calls:

CKLCS2, CRSRTR.

3.4.9 Subroutine EARWND(N1,ROWK)

1. Description:

EARWND scans active resources in RAVAIL for the window with the earliest start time as next window to be scheduled (WON, WOFF) in row ROWK.

2. Input Variable Names:

N1, RESAI(N1), RESW(2.N1).

3. Output Variable Names:

WON, WOFF, ROWK.

4. Called by:

MAIN.

5. Calls:

None.

3.4.10 Subroutine FILRD(NPOS)

1. Description:

FILRD reads a computed number of disk file records NOREC into the DIN array according to file position pointer NPOS, and passes the initial window read back in COMMON. Subsequent calls return the next window back until it becomes necessary to read again, or until end of mission time, or until end of file.

2. Input Variable Names:

BLKPTR, SUBPTR, NPOS, SIZ, IFILE(2), FLAG = ON/OFF code indicator, IBUF(32), NOREC, MSTART, MSTOP, TMIN, RDEND.

3. Output Variable Names:

BLKPTR, SUBPTR, NPOS, DIN(10000), RFLG, K, TON, TOFF.

4. Called by:

RESRD, CKLOS, CKIOS2.

5. Calls:

SEARCH, POSNSA RPOSSA, EXIT.

3.4.11 Subroutine INTRPT(TI,N1,N2)

1. Description:

INTRPT reads a complete set of each of resource setup files and requestor setup files at the interrupt/restart time which was a keyboard entry by the operator in MAIN, to re-parameterize resources and requestors. The order and number of resources and requestors must be maintained for each interrupt. Activated resource and requestor files are positioned by POSRES and POSREQ. INTRPT inputs the next interrupt time from the operator.

2. Input Variable Names:

N1, N2.

3. Output Variable Names:

INTFI., TI, I.

4. Called by:

MAIN.

5. Calls:

RSREAD, RQREAD, POSRES, POSREQ.

3.4.12 Subroutine MSG1(I,ROWK,T0)

1. Description:

MSG1 sends a scheduled request in RQTBL row I on the resource in RAVAIL row ROWK for the output disk file. Updates RQTBL row I by calling REQGEN.

2. Input Variable Names:

I, ROWK, T0, RESID(ROWK), REQID(I), REQPR(I), TN, REQW(2,I), REQPAR(10,I), RESPAR(10,ROWK), MSTP.

3. Output Variable Names:

BUFFER(10), TD(I), TSCH, TREQ, TAVI, IMSG1.

4. Called by:

SCHEDL.

5. Calls:

STOREQ, REQGEN.

3.4.13 Subroutine MSG2(I)

1. Description:

MSG2 sends an unfilled request message for the request in row I of RQTBL to the output queue for subsequent output to disk file. Updates the request in RQTBL by calling REQGEN.

2. Input Variable Names:

I, REQID(I), REQPR(I), REQW(2,I), REQPAR(10,I) TBEGIN, TEND, MSTP.

3. Output Variable Names:

BUFFER(10), IMSG2, TUNF, TREQ.

4. Called by:

REQUPD.

5. Calls:

STOREQ, REQGEN.

3.4.14 Subroutine MSG3(ROWK,TB,TE)

1. Description:

MSG3 sends an unused resource message for the resource in row ROWK of RAVAIL to the output queue for subsequent output to disk file.

2. Input Variable Names:

ROWK, RESID(ROWK), TB, TE, RESPAR(10,ROWK), MSTP.

3. Output Variable Names:

BUFFER(10), IMSG3, TWST, TAVL.

4. Called by:

SCHEDL, MAIN.

5. Calls:

STOREQ.

3.4.15 Subroutine OPTSCH(STATUS,ROWK,N2)

1. Description:

OPTSCH attempts to make an optimum relay schedule from all active requestors in RQTBL, according to priority, to start at beginning of resource window (WON,WOFF). See Appendix B.⁴ for a description of request generation algorithms.

2. Input Variable Names:

I, ROWK, N2, REQA(I), REQPR(I), CHECK, REQPAR(10,I),
REQW(2,I), WON, WOFF.

3. Output Variable Names:

T1, T2, I, ROWK, STATUS.

4. Called by:

SCHEDL.

5. Calls:

CKLOS.

3.4.16 Subroutine POSREQ(TI,I)

1. Description:

POSREQ determines if an active requestor window is positioned properly, in reference to the interrupt/restart time, to make it a schedule candidate. The window must allow the entire requested relay time to be available after the operator input interrupt time. Otherwise, another request window is generated until the position is proper.

2. Input Variable Names:

TI, I, REQA(I), REQW(2,I), REQPAR(6,I).

3. Output Variable Names.

REQW(2,I).

4. Called by:

INTRPT.

5. Calls:

REQGEN.

3.4.17 Subroutine POSRES(TI,I)

1. Description:

POSRES determines if an active resource available window is positioned properly in reference to the interrupt/restart time, to make it a schedule candidate. If the start of the window is beyond the operator input interrupt time, it is properly positioned. Otherwise, if the end of the window is beyond the interrupt time, the window start time is replaced by the interrupt time, shortening the available time for a proper fit. If neither, the next resource window is read in until the position is proper.

2. Input Variable Names:

TI, I, RESAI(I), RESW(2,I).

3. Output Variable Names:

RESW(2,I).

4. Called by:

INTRPT.

5. Calls:

RESRD.

3.4.18 Subroutine PRIORT(N2)

1. Description:

PRIORT computes current priority ratings for RQTBL requests. If a requestor is active and $WON \geq TR$, priority is set to 1. Otherwise, requestor priority is set to 2.

2. Input Variable Names:

N2, REQAI(N2), TR(N2), WON.

3. Output Variable Names

REQPR(N2).

4. Called by:

MAIN.

5. Calls:

None:

3.4.19 Subroutine REQGEN(I)

1. Description:

REQGEN generates various types of request requirements based on the request type code and request description parameters, which may designate a variable request time duration. The type code is used to select the appropriate algorithm to generate requests based on the accompanying parameters. Dependent IDs are requestor types 4 or 6 only. Refer to Appendix B.6 for algorithm definitions for all requestor types. REQGEN places the generated request requirements into RQTBL in the appropriate row. See Table 3-2.

2. Input Variable Names:

REQDID(5,I) NIDRQ(I), DIDRQ(5,I), REQAI(I), REQPAR(6,I), R(I), MSTART, TMIN, DELAY(4), MSTOP, 1.

3. Output Variable Names:

TR(I), TD(I), REQW(2,I).

4. Called by:

MAIN, MSG1, MSG2, POSREQ.

5. Calls:

EXIT.

3.4.20 Subroutine REQUPD(N2)

1. Description:

REQUPD replaces an outdated request in RQTBL. If current resource window (WON, WOFF) opens too late to satisfy a request, an unfilled request message is sent to the output queue, and a new request is generated.

2. Input Variable Names:

REQAI(N2), N2, WON, REQPAR(6,N2).

3. Output Variable Names:

I, REQW(2,I).

4. Called by:

MAIN.

5. Calls:

MSG2.

3.4.21 Subroutine RESRD(I)

1. Description:

RESRD places the next resource available window from FILRD into row I of RAVAIL array until end of file. After physical end of file or MSTOP time is reached, the resource is marked inactive.

2. Input Variable Names:

I, SUBPTR, BLKPTR, RESFN(2,I), NPOS, RFLG, RESID(I),
NK = dynamic pointer for DIN array, DIN(NK), DIN(NK+1).

3. Output Variable Names:

RESSUB(I), RESPTR(I), RESPOS(I), RESAI(I), IFILE(2),
RESW(2,I).

4. Called by:

MAIN, UPDRES, POSRES.

5. Calls:

FILRD, EXIT.

3.4.22 Subroutine RQREAD(N2)

1. Description:

RQREAD reads a manually generated requestor setup file according to the filename entered by the operator. The file is read into RQTBL requestor control arrays which specify and parameterize request requirements. Refer to Table 3-2 for a description of setup files.

2. Input Variable Names:

SETREQ(2,NNN), REQID(II), REQAI(II), REQFN(2,II),
REQDS(5,II), REQPR(II), REQPAR(6,II), RQRSFN(10,II), REQDID(5,II),
DELAY(4,II), NOREQ.

where NNN = requestor setup file array index.
II = requestor row number in RQTBL.

3. Output Variable Names:

REQID(II), REQAI(II), REQFN(2,II), REQDS(5,II), REQPR(II),
REQPAR(6,II), RQRSFN(10,II), REQDID(5,II), DELAY(4,II), N2, IN2, N.

5. Called by:

MAIN, INTRPT.

6. Calls:

SEARCH, EXIT.

3.4.23 Subroutine RQROW(I1)

1. Description:

RQROW determines the corresponding row number in RQTBL control array for requestor NQ in the output queue.

2. Input Variable Names:

NRQ, REQID(NRQ), NQ, QUEUE(14,NQ).

3. Output Variable Names:

I1.

4. Called by:

STATSV.

5. Calls.

None.

3.4.24 Subroutine RSREAD(N1)

1. Description:

RSREAD reads a manually generated resource setup file according to the filename entered by the operator. The file is read into RAVAIL resource control arrays which specify resource parameters. Refer to Table 3-2 for a description of setup files.

2. Input Variable Names:

SETRES(2,NN), RESID(I), RESAI(I), RESFN(2,I), RESDS(5,I), RESPR(I), RESPAR(6,I), NORES.

where NN = resource setup file array index.

I = resource row number in RAVAIL.

3. Output Variable Names:

N1, RESID(I), RESAI(I), RESFN(2,I), RESDS(5,I), RESPR(I), RESPAR(6,I), IN1, N.

4. Called by:

MAIN, INTRPT.

5. Calls:

SEARCH, EXIT.

3.4.25 Subroutine SCHEDL(ROWK,N2)

1. Description:

SCHEDL calls OPTSCH to attempt an optimum schedule for resource window (WON, WOFF) of RAVAIL from the active requests in RQTBL. If an optimum schedule is made, puts request row number I into STATUS, updates TN end time to WON+DTIME, sends scheduled request message to output queue, updates RQTBL, and finally updates resource in ROWK of RAVAIL.

If optimum schedule was not possible, calls EARSCH attempting to schedule resource in ROWK of RAVAIL with the earliest request that fits by priority. If resource can be scheduled by EARSCH, sets requestor row number I to STATUS, sends unused resource message for (WON,T1) of resource K, updates TE end time to BINT + DTIME, sends scheduled request message, updates RQTBL, and finally updates resource in RAVAIL.

If the entire resource cannot be scheduled, sends unused resource message to the output queue, saves resource for other requestor scheduling attempts, and updates RAVAIL.

2. Input Variable Names:

ROWK, N2, WON, REQPAR(10,I).

3. Output Variable Names:

I = scheduled request row number, TN, WON, WOFF, TE, BINT, STATUS.

4. Called by:

MAIN.

5. Calls:

OPTSCH, MSG1, UPDRES, EARSCH, MSG3, EXIT.

3.4.26 Subroutine STATSV

1. Description:

STATSV computes the statistics for generated class intervals. If INIT flag is zero, sets INIT to 1, generates next class interval and resets statistical arrays to zero. Otherwise if time of last output queue entry is greater than class interval, prints statistics of that interval by calling CLSSPR, which resets INIT to zero. If INIT is not zero, computes statistics for message types 1 and 2.

2. Input Variable Names:

INIT, CLASS, QUEUE(1,NQ), NQ, I1, QUEUE(10,NQ), MSTART, NRQ.

3. Output Variable Names:

NCD(I), NUNF(I), TIMSCD(I), REQUNF(I).

4. Called by:

WRITEQ.

5. Calls:

RQROW, CLSSPR.

3.4.27 Subroutine STOREQ

1. Description:

STOREQ moves message types 1,2,3 records from the output buffer area to the output queue. Tests for over-writing output queue boundary. Also converts time in minutes to seconds and milliseconds.

2. Input Variable Names:

NQ, BUFFER(10).

3. Output Variable Names:

QUEUE(32,NQ), ISECS, MSECS.

4. Called by:

MSG1, MSG2, MSG3.

5. Calls:

EXIT.

3.4.28 Subroutine TOTLPR

1. Description:

TOTLPR prints the total class interval statistics after either all resources or requestors are inactive, or statistical stop time is reached. Statistics include total number of unfilled dumps, total minutes lost, total number of scheduled dumps, and total minutes scheduled for each requestor.

2. Input Variable Names:

NUNFT(NRQ), NCDT(NRQ), RQUNFT(NRQ), TIMSDT(NRQ),
NRQ.

3. Output Variable Names:

NUNFT(NRQ), NCDT(NRQ), RQUNFT(NRQ), TIMSDT(NRQ).

4. Called by:

MAIN.

5. Calls:

EXIT.

3.4.29 Subroutine UPDRES(ROWK,TB,TE)

1. Description:

UPDRES updates resource by reducing resource window in ROWK of RAVAIL by (TB,TE). If resource is depleted, replaces it with next window from resource time line file by calling RESRD. If end of file is reached or MSTOP exceeded, marks resource inactive.

2. Input Variable Names:

ROWK, TE, WON, WOFF, RESW(2,ROWK).

3. Output Variable Names:

WON, RESW(1,ROWK).

4. Called by:

SCHEDL.

5. Calls:

RESRD.

3.4.30 Subroutine WRITEQ

1. Description:

WRITEQ sorts the output queue in descending order by calling CRSRTR, and writes all records in the queue with time \leq QTIME to the output disk file. Calls STATSV for statistical class interval computation and printing.

2. Input Variable Names:

NQ, QUEUE(32,NQ), WON.

3. Output Variable Names:

NREC, KEYS, NKEYS, NWD, QUEUE(32,NQ), IQUEUE(19).

4. Called by:

MAIN, CLEARQ.

5. Calls:

CRSRTR, STATSV, EXIT.

APPENDIX A
LANDSAT SCHEDULER

APPENDIX A.1 SAMPLE PRINTOUT OF THE LANDSAT SCHEDULER PROGRAM EXECUTION

OK. SIC-GLAND.DDPP
GO

ENTER CURRENT RUN DATE AND TIME
(FORMAT = MM.DD.YY.HH.MM)

5.2.88.3.43

ENTER OPERATORS NAME
(MAX. = 20 ALPHA-NUM CHARS)

JANET REEDER

ENTER INAME -- INPUT FILE NAME
(MAX. = 6 ALPHA-NUM CHARS)

LS001

ENTER ONAME -- OUTPUT FILE NAME
(MAX. = 6 ALPHA-NUM CHARS)

OC1001

ENTER REQID11 -- REQUEST ID FOR LANDSAT D (INTEGER)
(MAX. FORMAT = XXXX)

1

ENTER REQID12 -- REQUEST ID FOR LANDSAT-D PRIME (INTEGER)
(MAX. FORMAT = XXXX)

2

ENTER REQID13 -- REQUEST ID FOR PM TIME (INTEGER)
(MAX. FORMAT = XXXX)

22

ENTER RESID -- RESOURCE ID CODE (INTEGER)
(MAX. FORMAT = XXXX)

1

ENTER RECTYP -- REQUEST TYPE CODE (INTEGER)
(MAX. FORMAT = XXXX)

1

ENTER INSTANT -- MISSION START TIME IN HRS (REAL)
(MAX. FORMAT = XXXXXXXX.XXXXXX)

8.

NOTE:
OPERATOR INPUT UNDERLINED.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX A.1. (Continued)

ENTER TORB -- ORBITAL PERIOD IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
98.94	---
ENTER OTIME -- DURATION TIME OF R/T RELAY IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
18.	---
ENTER TOFFST -- OFFSET TIME BETWEEN LANDSAT 0 & 0 PRIME IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
37.679	---
ENTER OLTAT1 -- START OF POTENTIAL SCHEDULING PERIOD IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
0.	---
ENTER OLTAT2 -- LENGTH OF ALL POTENTIAL SCHEDULING PERIODS IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
49.46	---
ENTER PMSTRT -- START TIME OF THE FIRST PM PERIOD IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
0.	---
ENTER PMDUR -- DURATION TIME OF EACH PM PERIOD IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
68.	---
ENTER PMFREQ -- TIME BETWEEN SUCCESSIVE PM PERIODS IN MINS (REAL)	---
(MAX. FORMAT = XXXXXXXX.XXXXXX)	---
988.	---

APPENDIX A.1. (Continued)

CURRENT RUN DATE AND TIME	5/ 2/80	3:43	OPERATOR: JAKEL MCCORD
1. (I NAME)	INPUT FILE	=	LST041
2. (O NAME)	OUTPUT FILE	=	MCI041
3. (REQID1)	REQUEST ID CODE FOR LANDSAT C	=	1
4. (REQID2)	REQUEST ID CODE FOR LANDSAT D PRIME	=	2
5. (REQID3)	REQUEST ID CODE FOR PM TIME	=	99
6. (RESID)	RESOURCE ID CODE	=	1
7. (REQTYP)	REQUEST TYPE CODE	=	1
8. (MSTART)	MISSION START TIME (MINS)	=	0.000000
9. (MSTOP)	MISSION STOP TIME (MINS)	=	1500.000000
10. (TORR)	LANDSAT ORBITAL PERIOD (MINS)	=	98.93927
11. (OTIME)	DURATION OF R/T RELAY (MINS)	=	10.000000
12. (TOFFST)	OFFSET TIME BETWEEN LANDSAT D AND D PRIME (MINS)	=	37.678993
13. (DLTAT1)	START OF POTENTIAL SCHEDULING PERIOD (MINS)	=	0.000000
14. (DLTAT2)	LENGTH OF ALL POTENTIAL SCHEDULING PERIOD (MINS)	=	49.459999
15. (PMSTRT)	START TIME OF FIRST PM PERIOD (MINS)	=	0.000000
16. (PMOUR)	DURATION OF EACH PM PERIOD (MINS)	=	60.000000
17. (PMFREQ)	TIME BETWEEN SUCCESSIVE PM PERIODS (MINS)	=	400.000000

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX A.1. (Continued)

ORNO	REGORB	ENDORB	TOPS LINE-OF-SIGHT			SCHEDULE WINDOW			SCHEDULE		
			WON	WOFF	TON	TOFF	TB	TF			
1.	0.000	49.460	76.000	287.000	76.000	49.460	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
2.	98.940	148.400	76.000	287.000	98.940	148.400	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
3.	197.880	247.340	76.000	287.000	197.880	247.340	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
4.	296.820	346.280	318.000	384.000	318.000	346.280	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
5.	395.760	445.220	422.000	480.000	422.000	445.220	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
6.	494.700	544.160	540.000	583.000	540.000	544.160	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
7.	593.640	643.100	623.000	684.000	623.000	643.100	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
8.	692.580	742.040	721.000	792.000	721.000	742.040	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
9.	791.520	840.980	815.000	960.000	815.000	840.980	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
10.	890.460	939.920	815.000	960.000	890.460	939.920	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
11.	989.400	1038.860	1063.000	1126.000	1063.000	1038.860	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
12.	1088.340	1137.800	1063.000	1126.000	1088.340	1126.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
13.	1187.280	1236.740	1165.000	1225.000	1187.280	1225.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
14.	1286.220	1335.680	1266.000	1326.000	1286.220	1326.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
15.	1385.160	1434.620	1365.000	1428.000	1385.160	1428.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000	-4000000.000
END OF FILE			RFL6 = 1								

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX A.1. (Concluded)

```

MC1041  RUN SUMMARY STATISTICS AS OF  5/ 2/80  3:45

TOTAL RUN TIME (IN MINS)      1500.000
PREVENTATIVE MAINTENANCE TIME (IN MINS)  240.000
REMAINING TIME AVAILABLE (IN MINS)  1260.000
                                     16.02 **
                                     84.02 **

NO. OF REGS.      %      TIME (MINS.)      %
-----
FILLED REGS      21.000      70.0      378.000      30.0 *
UNFILLED REGS      9.000      30.0      162.000      12.9 *

TOTAL REGS      30.000

TOTAL TIME NOT SCHEDULED (EXCL. PM TIME)  882.000      70.0 *
TOTAL TIME NOT SCHEDULED (INCL. PM TIME)  1122.000      74.8 **
  
```

NOTE: * BASED ON REMAINING TIME AVAILABLE.
 ** BASED ON TOTAL RUN TIME.

OK-COMO-END

APPENDIX A.2 LANDSAT SCHEDULED SAMPLE OF THE OUTPUT FILE (NEW OUTPUT)

TIME IN SECONDS	TIME IN MINUTES	ON/OPS CODE	TYPE CODE	MSO CODE	REC TYPE	REC CODE	PRIOR RATING	REC ID	DEL-1A1	DEL-1A2
0	0.0000	1	0	0	0	0	0	1	0.0000	0.0000
0	0.0000	2	0	0	0	0	0	1	0.0000	0.0000
3600	60.0000	1	0	0	0	0	0	99	0.0000	0.0000
6240	104.0000	2	0	0	0	0	0	99	0.0000	0.0000
7200	120.0000	1	0	0	0	0	0	1	0.0000	0.0000
8599	143.3333	2	0	0	0	0	0	1	0.0000	0.0000
9599	161.6667	2	0	0	0	0	0	2	0.0000	0.0000
10599	180.0000	1	0	0	0	0	0	2	0.0000	0.0000
12333	205.5556	2	0	0	0	0	0	1	0.0000	0.0000
14013	233.5556	1	0	0	0	0	0	1	0.0000	0.0000
15594	259.9000	2	0	0	0	0	0	2	0.0000	0.0000
16674	277.9000	1	0	0	0	0	0	2	0.0000	0.0000
18249	304.1500	2	0	0	0	0	0	1	0.0000	0.0000
19249	320.8000	1	0	0	0	0	0	1	0.0000	0.0000
21330	355.5000	2	0	0	0	0	0	2	0.0000	0.0000
22610	376.8333	1	0	0	0	0	0	2	0.0000	0.0000
23407	390.1000	2	0	0	0	0	0	1	0.0000	0.0000
24567	409.4333	1	0	0	0	0	0	1	0.0000	0.0000
26808	446.8000	2	0	0	0	0	0	99	0.0000	0.0000
32400	540.0000	1	0	0	0	0	0	99	0.0000	0.0000
37401	623.3333	2	0	0	0	0	0	1	0.0000	0.0000
38521	642.0000	1	0	0	0	0	0	1	0.0000	0.0000
39702	661.7000	2	0	0	0	0	0	2	0.0000	0.0000
40792	679.8000	1	0	0	0	0	0	2	0.0000	0.0000
43377	722.9500	2	0	0	0	0	0	1	0.0000	0.0000
44957	749.9000	1	0	0	0	0	0	1	0.0000	0.0000
45630	760.5000	2	0	0	0	0	0	2	0.0000	0.0000
46710	778.5000	1	0	0	0	0	0	2	0.0000	0.0000
49240	820.7000	2	0	0	0	0	0	1	0.0000	0.0000
50320	838.7000	1	0	0	0	0	0	1	0.0000	0.0000
51500	858.0000	2	0	0	0	0	0	2	0.0000	0.0000
52580	876.6667	1	0	0	0	0	0	2	0.0000	0.0000
53184	886.4000	2	0	0	0	0	0	1	0.0000	0.0000
54264	905.0667	1	0	0	0	0	0	1	0.0000	0.0000
57670	961.1667	2	0	0	0	0	0	99	0.0000	0.0000
61230	1020.5000	1	0	0	0	0	0	99	0.0000	0.0000
63497	1058.2833	2	0	0	0	0	0	1	0.0000	0.0000
64577	1075.4167	1	0	0	0	0	0	1	0.0000	0.0000
67750	1129.1667	2	0	0	0	0	0	2	0.0000	0.0000
68030	1135.2500	1	0	0	0	0	0	2	0.0000	0.0000
71262	1187.8333	2	0	0	0	0	0	1	0.0000	0.0000
72362	1206.0333	1	0	0	0	0	0	1	0.0000	0.0000
73543	1225.7167	2	0	0	0	0	0	2	0.0000	0.0000
74623	1243.7167	1	0	0	0	0	0	2	0.0000	0.0000
77219	1286.9833	2	0	0	0	0	0	1	0.0000	0.0000
78299	1304.9833	1	0	0	0	0	0	1	0.0000	0.0000
79400	1323.0000	2	0	0	0	0	0	2	0.0000	0.0000
80540	1341.0000	1	0	0	0	0	0	2	0.0000	0.0000
83517	1391.9500	2	0	0	0	0	0	1	0.0000	0.0000
84597	1410.0000	1	0	0	0	0	0	1	0.0000	0.0000
86440	1440.0000	2	0	0	0	0	0	99	0.0000	0.0000
90000	1500.0000	1	0	0	0	0	0	99	0.0000	0.0000
90000	1500.0000	2	0	0	0	0	0	0	0.0000	0.0000

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX A.3 LANDSAT SCHEDULER ERROR MESSAGES

Following is a list of the possible program error messages that may appear while executing the Landsat Scheduler programs and the possible corrective action that should be taken:

ROUTINE	ERROR MESSAGE	EXPLANATION	CORRECTIVE ACTION
Main	'OPEN OR CLOSE FILE ERROR'	Error on opening or closing the input or output disk file. Program will exit, closing all open files, and will return control to the system.	<ul style="list-style-type: none"> -If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Check the input file name to be sure it matches the one in the UFD you are using. -Restart program execution beginning with Operator Instruction 2.
Main, WRITE	'WRITE ERROR ON UNIT 2'	Unit 2 refers to the output disk file. Program will exit, closing all open files, and will return control to the system.	<ul style="list-style-type: none"> -If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Check to see if the disk unit is full or has other problem. -Restart program execution beginning with Operator Instruction 2.
Main	'INPUT PARAMETER READ ERROR'	In responding to the CRT prompts for setup parameters, you used an invalid format. Program will exit, closing all open files, and will return control to the system.	<ul style="list-style-type: none"> -If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Check Figure 2-1 for correct parameter setup formats. -Restart program execution beginning with Operator Instruction 2.

APPENDIX A.3. (Concluded)

ROUTINE	ERROR MESSAGE	EXPLANATION	CORRECTIVE ACTION
DREAD	'PHYSICAL READ ERROR RFLG = 4'	While reading the input file from disk, you encountered a physical read error. Program will exit, closing all open files, and will return control to the system.	-If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Checkout the input file.
DREAD	'WINDOW ERROR RFLG = 3'	While reading the input file from disk, a record was encountered that did not contain a legitimate "on" or "off" code. Program will exit, closing all open files, and will return control to the system.	-If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Checkout the input file.
DREAD	'FILE CONTAINS TWO CONSECUTIVE END-OF-LOS WINDOWS RFLG = 5'	While reading the input file from disk, encountered two consecutive End-of-LOS Window codes. Program will exit, closing all open files, and will return control to the system.	-If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Input file should be corrected.
DREAD	'FILE CONTAINS TWO CONSECUTIVE BEGINNING-OF-LOS WINDOWS RFLG = 6'	While reading the input file from disk, encountered two consecutive Beginning-of-LOS Window codes. Program will exit, closing all open files, and will return control to the system.	-If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. -Input file should be corrected.

APPENDIX B
TDRS SCHEDULER

OK. SEG BYDORS. SCHED

[illegible]

REQUESTOR	S-TOP	FILENAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
REQUESTOR	S-TOP	FILENAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
REQUESTOR	S-TOP	FILENAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
REQUESTOR	S-TOP	FILENAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
REQUESTOR	S-TOP	FILENAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96				

SCHEDULING IN PROGRESS						
NO. WINDOWS PER FILE READ = 200						
(0.. 500.)	REQUESTOR 1	2	3	4	5
		# UNFILL DUMPS	0.0	0.0		
		# SCHED DUMPS	4.0	0.0		
		DUMP MINS LOST	0.0	0.0		
		DUMP MINS SCHED	50.0	0.0		
(500.. 1000.)		25.0	0.0		
		# UNFILL DUMPS	0.0	0.0		
		# SCHED DUMPS	3.0	1.0		
		DUMP MINS LOST	0.0	0.0		
		DUMP MINS SCHED	42.0	28.0		

NOTE: Ordinate (Y) of "mean" is 100.

APPENDIX B.1. (Concluded)

OK, SEG #IDRS.SCHED PAGE 0002

```

( 1000., 1500.)
# UNFILL DUMPS      0.0      0.0      0.0
# SCHED DUMPS      4.0      1.0      1.0
DUMP MINS LOST      0.0      0.0      0.0
CUMP MINS SCHED     56.0     20.0     20.0

TOTAL # UNFILL DUMPS      0.0      0.0      0.0
TOTAL # SCHED DUMPS     11.0      4.0      4.0
TOTAL MINS LOST          3.0      0.0      0.0
TOTAL MINS SCHED       154.0     20.0     20.0
  
```

PAUSE, HIT RETURN TO CONTINUE

```

TREQ(MINS)= 314.000      TUNF(MINS)= 0.000      TAVL(MINS)= 1991.999
TSCH(MINS)= 314.000      TOST(MINS)= 1678.000
TOTAL REQUESTS FILLED= 100%      TOTAL RESOURCE UTILIZATION= 15%
SCHEDULED REQUESTS = 1%      UNFILLED REQUESTS = 0      UNUSED RESOURCES = 66
  
```

OK, COMO -END

Example 1. Resource Setup File

[illegible]

Example 2. Requestor Setup File

[illegible]

76

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B.3. (Continued)

TIME IN	TIME IN	ON/OFF	MSG	REQ	RES	PRIOR	REQ	RES	DE PAGE
27033	450-5566	2	2	3	1	0	0	1	19-6588
28467	474-4666	2	2	3	1	0	0	2	28-1500
28713	478-5666	1	1	3	0	0	0	1	8-8888
29047	482-4666	1	1	3	0	0	0	1	8-8888
30726	512-1165	2	2	3	0	0	0	2	0-0000
31320	522-0000	2	2	3	0	0	0	2	12-6599
31826	530-1165	1	1	3	0	0	0	1	44-4333
32460	541-0000	1	1	3	0	0	0	1	0-0000
33006	550-1165	2	2	3	0	0	0	2	0-0000
33306	556-1165	2	2	3	0	0	0	2	20-0000
33859	561-4499	2	2	3	0	0	0	2	0-0000
34404	574-1165	2	2	3	0	0	0	2	19-6499
34718	578-5128	1	1	3	0	0	0	1	23-3000
35028	581-4165	1	1	3	0	0	0	1	8-8888
35558	592-4499	2	2	3	0	0	0	2	0-0000
36663	611-0665	2	2	3	0	0	0	2	14-0000
37443	624-0665	2	2	3	0	0	0	2	12-6530
37743	629-0665	1	1	3	0	0	0	1	31-4167
38043	632-0665	1	1	3	0	0	0	1	0-0000
38443	635-0665	2	2	3	0	0	0	2	0-0000
38743	638-0665	2	2	3	0	0	0	2	0-0000
39043	641-0665	2	2	3	0	0	0	2	0-0000
39343	644-0665	2	2	3	0	0	0	2	0-0000
39643	647-0665	2	2	3	0	0	0	2	0-0000
40043	650-0665	2	2	3	0	0	0	2	0-0000
40343	653-0665	2	2	3	0	0	0	2	0-0000
40643	656-0665	2	2	3	0	0	0	2	0-0000
40943	659-0665	2	2	3	0	0	0	2	0-0000
41243	662-0665	2	2	3	0	0	0	2	0-0000
41543	665-0665	2	2	3	0	0	0	2	0-0000
41843	668-0665	2	2	3	0	0	0	2	0-0000
42143	671-0665	2	2	3	0	0	0	2	0-0000
42443	674-0665	2	2	3	0	0	0	2	0-0000
42743	677-0665	2	2	3	0	0	0	2	0-0000
43043	680-0665	2	2	3	0	0	0	2	0-0000
43343	683-0665	2	2	3	0	0	0	2	0-0000
43643	686-0665	2	2	3	0	0	0	2	0-0000
43943	689-0665	2	2	3	0	0	0	2	0-0000
44243	692-0665	2	2	3	0	0	0	2	0-0000
44543	695-0665	2	2	3	0	0	0	2	0-0000
44843	698-0665	2	2	3	0	0	0	2	0-0000
45143	701-0665	2	2	3	0	0	0	2	0-0000
45443	704-0665	2	2	3	0	0	0	2	0-0000
45743	707-0665	2	2	3	0	0	0	2	0-0000
46043	710-0665	2	2	3	0	0	0	2	0-0000
46343	713-0665	2	2	3	0	0	0	2	0-0000
46643	716-0665	2	2	3	0	0	0	2	0-0000
46943	719-0665	2	2	3	0	0	0	2	0-0000
47243	722-0665	2	2	3	0	0	0	2	0-0000
47543	725-0665	2	2	3	0	0	0	2	0-0000
47843	728-0665	2	2	3	0	0	0	2	0-0000
48143	731-0665	2	2	3	0	0	0	2	0-0000
48443	734-0665	2	2	3	0	0	0	2	0-0000
48743	737-0665	2	2	3	0	0	0	2	0-0000
49043	740-0665	2	2	3	0	0	0	2	0-0000
49343	743-0665	2	2	3	0	0	0	2	0-0000
49643	746-0665	2	2	3	0	0	0	2	0-0000
49943	749-0665	2	2	3	0	0	0	2	0-0000
50243	752-0665	2	2	3	0	0	0	2	0-0000
50543	755-0665	2	2	3	0	0	0	2	0-0000
50843	758-0665	2	2	3	0	0	0	2	0-0000
51143	761-0665	2	2	3	0	0	0	2	0-0000
51443	764-0665	2	2	3	0	0	0	2	0-0000
51743	767-0665	2	2	3	0	0	0	2	0-0000
52043	770-0665	2	2	3	0	0	0	2	0-0000
52343	773-0665	2	2	3	0	0	0	2	0-0000
52643	776-0665	2	2	3	0	0	0	2	0-0000
52943	779-0665	2	2	3	0	0	0	2	0-0000
53243	782-0665	2	2	3	0	0	0	2	0-0000
53543	785-0665	2	2	3	0	0	0	2	0-0000
53843	788-0665	2	2	3	0	0	0	2	0-0000
54143	791-0665	2	2	3	0	0	0	2	0-0000
54443	794-0665	2	2	3	0	0	0	2	0-0000
54743	797-0665	2	2	3	0	0	0	2	0-0000
55043	800-0665	2	2	3	0	0	0	2	0-0000
55343	803-0665	2	2	3	0	0	0	2	0-0000
55643	806-0665	2	2	3	0	0	0	2	0-0000
55943	809-0665	2	2	3	0	0	0	2	0-0000
56243	812-0665	2	2	3	0	0	0	2	0-0000
56543	815-0665	2	2	3	0	0	0	2	0-0000
56843	818-0665	2	2	3	0	0	0	2	0-0000
57143	821-0665	2	2	3	0	0	0	2	0-0000
57443	824-0665	2	2	3	0	0	0	2	0-0000
57743	827-0665	2	2	3	0	0	0	2	0-0000
58043	830-0665	2	2	3	0	0	0	2	0-0000
58343	833-0665	2	2	3	0	0	0	2	0-0000
58643	836-0665	2	2	3	0	0	0	2	0-0000
58943	839-0665	2	2	3	0	0	0	2	0-0000
59243	842-0665	2	2	3	0	0	0	2	0-0000

APPENDIX B.4 TDRS SCHEDULER PROGRAM ERROR MESSAGES

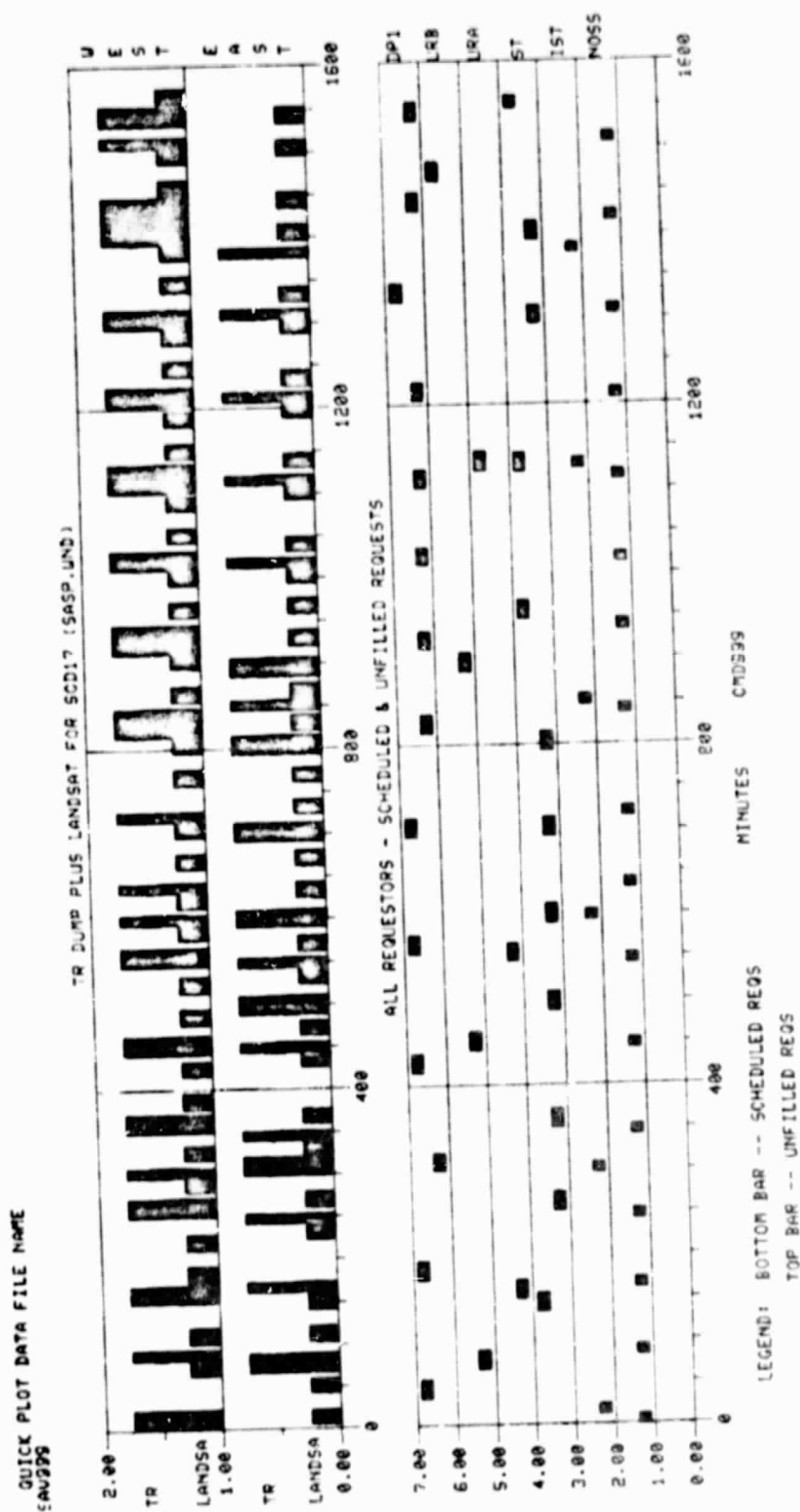
Following is a list of the possible program error messages that may appear while executing the TDRS Scheduler program and the possible corrective action that should be taken:

ROUTINE	ERROR MESSAGE	EXPLANATION	CORRECTIVE ACTION
MAIN	Error, duplicate request ID.	Must use unique ID.	Program closes requestor setup file and returns control to system monitor by 'CALL EXIT'.
MAIN	Error, duplicate resource ID.	Must use unique ID.	Program closes resource setup file and returns control to system monitor by 'CALL EXIT'.
STOREQ	Error, exceeded output queue boundary.	No. of status change messages exceeded program limit.	Program returns control to system monitor with 'CALL EXIT'.
FILRD	Error, inconsistent OFF code on File 'filename', Record '....'.	Break in ON/OFF code sequence for resource available window on scheduled time line file.	Operator must enter carriage return to continue.
FILRD	Error, inconsistent ON code on File 'filename', Record '....'.	Break in ON/OFF code sequence for resource available window on scheduled time line file.	Operator must enter carriage return to continue.
FILRD	Error, MSTOP reached but no ON-OFF code for res. file 'filename', Record '....'.	Mission stop time reached without ON-OFF code.	Program closes file and returns control to system monitor with 'CALL EXIT'.
MAIN	Error, no. of windows less than 1.	Program requires at least 1 resource available window to attempt data relay scheduling.	Program returns control to system monitor with 'CALL EXIT'.
FILRD	Error, no ON-OFF code for resource window on File 'filename', 'Record'....	No opening-closing code for resource available window on scheduled time line file.	Program closes file and returns control to system monitor with 'CALL EXIT'.

APPENDIX B.4. (Concluded)

ROUTINE	ERROR MESSAGE	EXPLANATION	CORRECTIVE ACTION
MAIN.FILRD	Error on opening/closing disk file.	File can be determined from routine where error occurred.	Program returns control to system monitor with 'CALL EXIT'.
MAIN. CLSSPR. FILRD. REQGEN. RESRD. SCHEDL. STOREQ. TOTLPR. WRITEQ	Error on read/write to CRT.	Self explanatory.	Program returns control to system monitor with 'CALL EXIT'.
MAIN.FILRD WRITEQ	Error on read/write to disk file.	File can be determined from routine where error occurred.	Program returns control to system monitor with 'CALL EXIT'.
MAIN	Exceeded request dimension.	Program allows maximum 25 requestors.	Program returns control to system monitor with 'CALL EXIT'.
MAIN	Exceeded resource dimension.	Programs allows maximum 10 resources.	Program returns control to system monitor by 'CALL EXIT'.
REQGEN	Invalid request type.	Request type from requestor set-up file must be either 3.4.5.6. or 7.	Program returns control to system monitor with 'CALL EXIT'.
MAIN	Total requests filled is undefined - division by zero.	To compute the percentage of total requests filled, program divides by total requests.	Program continues.
MAIN	Total resource utiliz. undefined - division by zero.	To compute the percentage of total resource utilization, program divides by total resources available.	Program continues.

APPENDIX B.5 SAMPLE OUTPUT FROM THE AREAEXPAND PROGRAM



ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B.6 REQUEST ALGORITHM GENERATION

REQGEN generates various types of request requirements based on the request type code and request description parameters. The type code determines which requestor algorithm to use. REQGEN places the generated request requirements into RQTBL in the appropriate row I.

B.6.1 TYPES 3 and 5

For request type codes 3 or 5, REQGEN generates a variably positioned request window of fixed width, once each n orbits. The window is defined by $(TR(I)+k1, TR(I)+k2)$, where $TR(I)$ is a reference time initially generated randomly within the first n orbits. To generate initial window reference time $TR(I)$,

$$TR(I) = R(I) * \text{orbital period}(I) * n(I) + \text{MSTART}.$$

See Table 3.2 and the TDRS Scheduler Subroutines, Section 3.4, for control array names and program variable definitions.

If there are dependent requestors (types 4 and 6 only), for this parent requestor, REQGEN sets up $TD(K)$ reference time for the dependent IDs.

$$TD(K) = (-\text{orbital period}(I) * n(K)) + TR(I) + \\ (\text{DELAY}(J) / 360) * \text{orbital period}(I)$$

where J = index number of dependent ID and K = row number in RQTBL for dependent ID J. Note $TR(K)$ must be initialized to 0.

The updated reference time for types 3 and 5 is computed by

$$TR(I) = TD(I) + \text{orbital period}(I) * n(I)$$

Then to generate the request window,

$$REQW(1,I) = TR(I) + k1(I)$$

$$REQW(2,I) = TR(I) + k2(I)$$

REQGEN checks for a variable length relay time if 1000 was added to the request type code in the setup file. Variable length data relay time periods are for all requestor types.

REQPAR(7,I)=RMEAN=PPAR(1)

REQPAR(8,I)=RMIN=PPAR(2)

REQPAR(9,I)=RMAX=PPAR(3)

REQPAR(10,I)=SIGMA=PPAR(4)

The function RNORM(PPAR,ISTRM) generates a relay time that varies randomly between RMIN and RMAX with mean and standard deviation of RMEAN and SIGMA. ISTRM is the stream number for the DRAND function which is invoked by RNORM.

Real-time variable length relays are simulated by loading

REQPAR(4,I) = REQPAR(5,I) ; i.e., $k_1 = k_2$.

REQGEN tests for this condition to correctly compute the request window start and stop times by using the current data relay period. For fixed length data relays, real-time requests are simulated by loading k_1+k_2 equal to the data relay period DTIME.

B.6.2 TYPES 4 and 6

For request types 4 and 6, REQGEN generates a request window, as for types 3 and 5, but for each m minutes rather than each n orbits. Thus to generate the initial window reference time,

$TR(I) = R(I) * m(I) + MSTART$.

Note $TR(I)$ must be preset to 0 in RQTBL row 1.

If there are dependent requestors for this parent requestor, REQGEN sets up $TD(K)$ reference time for the dependent IDs.

$TD(K) = m(K) + TR(I) + (DELAY(J)/360) * \text{orbital period}(I)$

The updated reference time for types 4 and 6 is computed by

$$TR(I) = TD(I) + m(I)$$

To generate the request window,

$$REQW(1,I) = TR(I) + k1(I)$$

$$REQW(2,I) = TR(I) + k2(I)$$

The variable length relay time and real-time variable length relays are as described in B.6.1.

B.6.3 TYPE 7

The initial window reference time for requestor type 7 is defined by

$$TR(I) = MSTART.$$

RQTBL will contain the parameters RTYPE,p,NBAR,k1,k2 and DTIME in REQPAR (1,I)...REQPAR(6,I), where RBAR is the mean time between successive request windows. The updated reference time for type 7 is computed by

$$TR(I) = TR(I) + R(I)$$

where $R(I) = EXPON(NBAR,I)$. The request window follows:

$$REQW(1,I) = TR(I) + k1(I)$$

$$REQW(2,I) = TR(I) + k2(I)$$

Type 7 may be real-time if k1 and k2 are chosen such that their difference is equal to DTIME. See B.6.1 for variable length relay time description.

APPENDIX C

FILESET

APPENDIX C.1
USER's GUIDE FOR FILESET

This program provides the user with the following options:

- A) Read or write a resource setup file
- B) Read or write a requestor setup file.

Both types of files can be used with the program #TDRS. SCHED (Refer to Table 3-2).

Procedures: To use the program FILESET, the following steps, some in response to the prompts on the video screen, should be taken:

- 1) To start program enter SEG #FILESET.ONE (return).

- 2) Respond to prompts:

"Enter R/W parameters IFLAG & INTL IFLG = 1 to read,
IFLAG = 2 to write NTL = number (1 to 10) of filenames...
Enter a negative value for..."

Enter 2 integer values for proper functions.

Ex. 2 2 (return)

- 3) Respond to I/O filename prompt by entering a 6 alphanumeric word.

Ex. TEST12 (return)

If name entered is not a new file, a prompt will instruct the user to enter a 1 if the old file can be deleted or some other number if not.

- 4) Respond to resource/requestor ID prompt. Enter proper integer value (up to 6 digits).

Ex. 2001 (return)

**At this point the program allows the user to cancel the program execution by entering a negative value.

- 5) Respond to active/inactive code prompt. Enter proper integer value.

Ex. 0 (return)

- 6) Respond to resource/requestor filename (up to 6 characters) by entering filename.

Ex. REQSTR (return)

7) Respond to file description prompt. Enter up to 20 characters.

8) Respond to priority code and 6 description parameters prompt. Enter proper integer/real values. If remaining parameters are zero, a / (slash) can be used as a delimiter.

Ex. 0 1 99.5 2/ (return)

The program has completed one cycle for resource setup file and is ready to start the second cycle (step 5). Enter negative value to quit (see **).

9) If type code is greater than 1000, respond to "Mean, Min, Max, and Sigma" prompt. Enter proper real values.

10) Respond to requestor/resource LOS filenames prompt. Enter proper filename (up to 6 characters) and hit return. Hit return twice after last entry.

Ex. LOS123 (return)

LOS234 (return)

(return)

11) Respond to ID parameter prompt. Enter proper integer values. Again if the remaining parameters are zero, a / (slash) can be used.

Ex. 0 1 / (return)

At this point the program has completed one cycle for the requestor setup file and is ready to start the second cycle (step 5). Enter negative value to quit (see **).

APPENDIX C.2
EXAMPLES OF EXECUTION RUNS

OK, SEG #FILESET.ONE

OK, SEG #FILESET.ONE
GO

ENTER R/W PARAMETERS IFLG & NTL
IFLG = 1 TO READ, IFLG = 2 TO WRITE
NTL = NUMBER, FROM 1 TO 10, OF FILE NAMES FOR LOSFILE ARRAY
ENTER A NEGATIVE VALUE FOR NTL TO READ OR WRITE A RESOURCE FILE
2 -2

ENTER 6 CHARACTER NAME FOR I/O FILE
SETRES

ENTER RESOURCE OR REQUESTER ID. ENTER A
NEGATIVE VALUE TO CANCEL PROGRAM
1

ENTER ACTIVE/INACTIVE CODE: 0=ACTIVE, 1=INACTIVE
0

ENTER RESOURCE OR REQUESTER FILENAME: UP TO 6 CHARS.
SECSZE

ENTER FILE DESCRIPTION TEXT: UP TO 20 CHARS.
TDRS EAST

ENTER PRIORITY CODE AND 6 DESCRIPTION PARAMETERS:
TYPE CODE,P(1),...,P(6)
0 1 /

ENTER RESOURCE OR REQUESTER ID. ENTER A
NEGATIVE VALUE TO CANCEL PROGRAM
2

ENTER ACTIVE/INACTIVE CODE: 0=ACTIVE, 1=INACTIVE.
0

ENTER RESOURCE OR REQUESTER FILENAME: UP TO 6 CHARS.
SECSZW

ENTER FILE DESCRIPTION TEXT: UP TO 20 CHARS.
TDRS WEST

ENTER PRIORITY CODE AND 6 DESCRIPTION PARAMETERS:
TYPE CODE,P(1),...,P(6)
0 1 /

ENTER RESOURCE OR REQUESTER ID. ENTER A
NEGATIVE VALUE TO CANCEL PROGRAM
-1

OK, COMO -END

NOTE: Operator input underlined.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C.2. (Continued)

OK. SEG #FILESET.CAF
GO

ENTER R/W PARAMETERS IFLG & NTL
IFLG = 1 TO READ, IFLG = 2 TO WRITE
NTL = NUMBER, FROM 1 TO 14, OF FILE NAMES FOR LOSFILE ARRAY
ENTER A NEGATIVE VALUE FOR NTL TO READ OR WRITE A RESOURCE FILE
2 3

ENTER 6 CHARACTER NAME FOR I/O FILE
SETREQ

ENTER RESOURCE OR REQUESTER ID. ENTER A
NEGATIVE VALUE TO CANCEL PROGRAM
8

ENTER ACTIVE/INACTIVE CODE: 0=ACTIVE, 1=INACTIVE

0

ENTER RESOURCE OR REQUESTER FILENAME: UP TO 6 CHARS.

COSE

ENTER FILE DESCRIPTION TEXT: UP TO 20 CHARS.

COSMIC DO EYE

ENTER PRIORITY CODE AND 6 DESCRIPTION PARAMETERS:

TYPE CODE, P(1), ..., P(6)

0 4 103. 240. -120. 100. 14.

ENTER REQUESTER/RESOURCE LINE OF SIGHT FILENAMES:

HIT CR AFTER EACH FILENAME AND TWICE

AFTER THE LAST NAME

COB041

COB171

ENTER DEPENDENT ID PARAMETERS:

NID _ NO. OF DEPENDENT REQUESTERS FOR THIS PARENT ID

DID(1), ..., DID(4) _ DEPENDENT ID ARRAY

DELAY(1), ..., (4) _ DELAY ARRAY (IN DEGREES)

0 7

ENTER RESOURCE OR REQUESTER ID. ENTER A
NEGATIVE VALUE TO CANCEL PROGRAM

9

ENTER ACTIVE/INACTIVE CODE: 0=ACTIVE, 1=INACTIVE

0

ENTER RESOURCE OR REQUESTER FILENAME: UP TO 6 CHARS.

ENTER FILE DESCRIPTION TEXT: UP TO 20 CHARS.

UABS-A

ENTER PRIORITY CODE AND 6 DESCRIPTION PARAMETERS:

TYPE CODE, P(1), ..., P(6)

0 4 103. 460. -120. 120. 20.

ENTER REQUESTER/RESOURCE LINE OF SIGHT FILENAMES:

HIT CR AFTER EACH FILENAME AND TWICE

AFTER THE LAST NAME

UAA041

UAA171

ENTER DEPENDENT ID PARAMETERS:

NID _ NO. OF DEPENDENT REQUESTERS FOR THIS PARENT ID

NOTE: Operator input underlined.

APPENDIX C.2. (Concluded)

OK, SEG #FILESET.ONE

DID(1).....DID(4) - DEPENDENT ID ARRAY

DELAY(1).....(4) - DELAY ARRAY (IN DEGREES)

1 10 0 0 0 180. /

ENTER RESOURCE OR REQUESTER ID. ENTER A

NEGATIVE VALUE TO CANCEL PROGRAM

10

ENTER ACTIVE/INACTIVE CODE: 0=ACTIVE, 1=INACTIVE

0

ENTER RESOURCE OR REQUESTER FILENAME: UP TO 6 CHARS.

ENTER FILE DESCRIPTION TEXT: UP TO 20 CHARS.

UARS-R

ENTER PRIORITY CODE AND 6 DESCRIPTION PARAMETERS:

TYPE CODE,P(1).....P(5)

0 6 103. 480. -120. 120. 20.

ENTER REQUESTER/RESOURCE LINE OF SIGHT FILENAMES:

HIT CR AFTER EACH FILENAME AND TWICE

AFTER THE LAST NAME

UAB041

UAB171

ENTER DEPENDENT ID PARAMETERS:

NID - NO. OF DEPENDENT REQUESTERS FOR THIS PARENT ID

DID(1).....DID(4) - DEPENDENT ID ARRAY

DELAY(1).....(4) - DELAY ARRAY (IN DEGREES)

0 /

ENTER RESOURCE OR REQUESTER ID. ENTER A

NEGATIVE VALUE TO CANCEL PROGRAM

-1

OK, COMO -END

NOTE: Operator input underlined.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX D

TL.GEN

APPENDIX D.1 GENERAL DESCRIPTION

TLGEN is an interactive, manual timeline generation program. It provides an output timeline file (32-word record line-of-sight windows) to be used by the Landsat Scheduler programs.

The CRT operator will be required to input only two words for each 32-word record. The first input word will be the record word 1 and will represent a start or stop time in seconds. The second input word will be the record word 16 and will represent either an "on" (1) code, or an "off" (2) code. (Note: 1 means the Landsat can be seen by the TDRS; 2 means the Landsat is no longer in sight of the TDRS.)

In order to create a valid file, the input times must be sequential, and the "on-off" codes must be alternating. There will be no limit to the number of records that may be input.

APPENDIX D.2 SAMPLE PRINTOUT OF THE TLGEN PROGRAM EXECUTION

This is a sample of an input file with CRT operator input annotated.

OK, OPEN SAMPLE 1
OK, END ATTACH
GO

0000 1
3000 0
6000 1
9000 2
12000 1
15000 2
18000 1
21000 0
24000 1
27000 2
30000 1
34000 2
37000 1
40000 0
42000 1
46000 2
50000 1
54000 2
58000 1
62000 0
66000 1
70000 0
74000 1
78000 2
82000 1
86000 0
90000 1
94000 2
98000 1
100000 0
104000 1
108000 2
112000 1
116000 0
120000 1
124000 2
128000 1
132000 0
136000 1
140000 2
144000 1
148000 0
152000 1
156000 2
160000 1
164000 0
168000 1
172000 2
176000 1
180000 0
184000 1
188000 2
192000 1
196000 0
200000 1
204000 2
208000 1
212000 0
216000 1
220000 2
224000 1
228000 0
232000 1
236000 2
240000 1
244000 0
248000 1
252000 2
256000 1
260000 0
264000 1
268000 2
272000 1
276000 0
280000 1
284000 2
288000 1
292000 0
296000 1
300000 2
304000 1
308000 0
312000 1
316000 2
320000 1
324000 0
328000 1
332000 2
336000 1
340000 0
344000 1
348000 2
352000 1
356000 0
360000 1
364000 2
368000 1
372000 0
376000 1
380000 2
384000 1
388000 0
392000 1
396000 2
400000 1
404000 0
408000 1
412000 2
416000 1
420000 0
424000 1
428000 2
432000 1
436000 0
440000 1
444000 2
448000 1
452000 0
456000 1
460000 2
464000 1
468000 0
472000 1
476000 2
480000 1
484000 0
488000 1
492000 2
496000 1
500000 0
504000 1
508000 2
512000 1
516000 0
520000 1
524000 2
528000 1
532000 0
536000 1
540000 2
544000 1
548000 0
552000 1
556000 2
560000 1
564000 0
568000 1
572000 2
576000 1
580000 0
584000 1
588000 2
592000 1
596000 0
600000 1
604000 2
608000 1
612000 0
616000 1
620000 2
624000 1
628000 0
632000 1
636000 2
640000 1
644000 0
648000 1
652000 2
656000 1
660000 0
664000 1
668000 2
672000 1
676000 0
680000 1
684000 2
688000 1
692000 0
696000 1
700000 2
704000 1
708000 0
712000 1
716000 2
720000 1
724000 0
728000 1
732000 2
736000 1
740000 0
744000 1
748000 2
752000 1
756000 0
760000 1
764000 2
768000 1
772000 0
776000 1
780000 2
784000 1
788000 0
792000 1
796000 2
800000 1
804000 0
808000 1
812000 2
816000 1
820000 0
824000 1
828000 2
832000 1
836000 0
840000 1
844000 2
848000 1
852000 0
856000 1
860000 2
864000 1
868000 0
872000 1
876000 2
880000 1
884000 0
888000 1
892000 2
896000 1
900000 0
904000 1
908000 2
912000 1
916000 0
920000 1
924000 2
928000 1
932000 0
936000 1
940000 2
944000 1
948000 0
952000 1
956000 2
960000 1
964000 0
968000 1
972000 2
976000 1
980000 0
984000 1
988000 2
992000 1
996000 0
1000000 1
1004000 2
1008000 1
1012000 0
1016000 1
1020000 2
1024000 1
1028000 0
1032000 1
1036000 2
1040000 1
1044000 0
1048000 1
1052000 2
1056000 1
1060000 0
1064000 1
1068000 2
1072000 1
1076000 0
1080000 1
1084000 2
1088000 1
1092000 0
1096000 1
1100000 2
1104000 1
1108000 0
1112000 1
1116000 2
1120000 1
1124000 0
1128000 1
1132000 2
1136000 1
1140000 0
1144000 1
1148000 2
1152000 1
1156000 0
1160000 1
1164000 2
1168000 1
1172000 0
1176000 1
1180000 2
1184000 1
1188000 0
1192000 1
1196000 2
1200000 1
1204000 0
1208000 1
1212000 2
1216000 1
1220000 0
1224000 1
1228000 2
1232000 1
1236000 0
1240000 1
1244000 2
1248000 1
1252000 0
1256000 1
1260000 2
1264000 1
1268000 0
1272000 1
1276000 2
1280000 1
1284000 0
1288000 1
1292000 2
1296000 1
1300000 0
1304000 1
1308000 2
1312000 1
1316000 0
1320000 1
1324000 2
1328000 1
1332000 0
1336000 1
1340000 2
1344000 1
1348000 0
1352000 1
1356000 2
1360000 1
1364000 0
1368000 1
1372000 2
1376000 1
1380000 0
1384000 1
1388000 2
1392000 1
1396000 0
1400000 1
1404000 2
1408000 1
1412000 0
1416000 1
1420000 2
1424000 1
1428000 0
1432000 1
1436000 2
1440000 1
1444000 0
1448000 1
1452000 2
1456000 1
1460000 0
1464000 1
1468000 2
1472000 1
1476000 0
1480000 1
1484000 2
1488000 1
1492000 0
1496000 1
1500000 2
1504000 1
1508000 0
1512000 1
1516000 2
1520000 1
1524000 0
1528000 1
1532000 2
1536000 1
1540000 0
1544000 1
1548000 2
1552000 1
1556000 0
1560000 1
1564000 2
1568000 1
1572000 0
1576000 1
1580000 2
1584000 1
1588000 0
1592000 1
1596000 2
1600000 1
1604000 0
1608000 1
1612000 2
1616000 1
1620000 0
1624000 1
1628000 2
1632000 1
1636000 0
1640000 1
1644000 2
1648000 1
1652000 0
1656000 1
1660000 2
1664000 1
1668000 0
1672000 1
1676000 2
1680000 1
1684000 0
1688000 1
1692000 2
1696000 1
1700000 0
1704000 1
1708000 2
1712000 1
1716000 0
1720000 1
1724000 2
1728000 1
1732000 0
1736000 1
1740000 2
1744000 1
1748000 0
1752000 1
1756000 2
1760000 1
1764000 0
1768000 1
1772000 2
1776000 1
1780000 0
1784000 1
1788000 2
1792000 1
1796000 0
1800000 1
1804000 2
1808000 1
1812000 0
1816000 1
1820000 2
1824000 1
1828000 0
1832000 1
1836000 2
1840000 1
1844000 0
1848000 1
1852000 2
1856000 1
1860000 0
1864000 1
1868000 2
1872000 1
1876000 0
1880000 1
1884000 2
1888000 1
1892000 0
1896000 1
1900000 2
1904000 1
1908000 0
1912000 1
1916000 2
1920000 1
1924000 0
1928000 1
1932000 2
1936000 1
1940000 0
1944000 1
1948000 2
1952000 1
1956000 0
1960000 1
1964000 2
1968000 1
1972000 0
1976000 1
1980000 2
1984000 1
1988000 0
1992000 1
1996000 2
2000000 1
2004000 0
2008000 1
2012000 2
2016000 1
2020000 0
2024000 1
2028000 2
2032000 1
2036000 0
2040000 1
2044000 2
2048000 1
2052000 0
2056000 1
2060000 2
2064000 1
2068000 0
2072000 1
2076000 2
2080000 1
2084000 0
2088000 1
2092000 2
2096000 1
2100000 0
2104000 1
2108000 2
2112000 1
2116000 0
2120000 1
2124000 2
2128000 1
2132000 0
2136000 1
2140000 2
2144000 1
2148000 0
2152000 1
2156000 2
2160000 1
2164000 0
2168000 1
2172000 2
2176000 1
2180000 0
2184000 1
2188000 2
2192000 1
2196000 0
2200000 1
2204000 2
2208000 1
2212000 0
2216000 1
2220000 2
2224000 1
2228000 0
2232000 1
2236000 2
2240000 1
2244000 0
2248000 1
2252000 2
2256000 1
2260000 0
2264000 1
2268000 2
2272000 1
2276000 0
2280000 1
2284000 2
2288000 1
2292000 0
2296000 1
2300000 2
2304000 1
2308000 0
2312000 1
2316000 2
2320000 1
2324000 0
2328000 1
2332000 2
2336000 1
2340000 0
2344000 1
2348000 2
2352000 1
2356000 0
2360000 1
2364000 2
2368000 1
2372000 0
2376000 1
2380000 2
2384000 1
2388000 0
2392000 1
2396000 2
2400000 1
2404000 0
2408000 1
2412000 2
2416000 1
2420000 0
2424000 1
2428000 2
2432000 1
2436000 0
2440000 1
2444000 2
2448000 1
2452000 0
2456000 1
2460000 2
2464000 1
2468000 0
2472000 1
2476000 2
2480000 1
2484000 0
2488000 1
2492000 2
2496000 1
2500000 0
2504000 1
2508000 2
2512000 1
2516000 0
2520000 1
2524000 2
2528000 1
2532000 0
2536000 1
2540000 2
2544000 1
2548000 0
2552000 1
2556000 2
2560000 1
2564000 0
2568000 1
2572000 2
2576000 1
2580000 0
2584000 1
2588000 2
2592000 1
2596000 0
2600000 1
260400

APPENDIX D.3 OPERATING INSTRUCTIONS

1) Sign in on the PRIME 400 using the appropriate user file directory (UFD) where the files to be used will be located.

2) For a hard copy of everything that appears on the CRT during a program execution run, enter 'COMO filename' for eventual spooling to the line printer.

3) Before the program can begin execution, the CRT operator must designate an output file on the disk by entering 'OPEN filename 1 2'. The filename may contain up to six alpha-numeric characters.

4) Begin execution by entering 'SEG #TLGEN'. System will respond with a 'GO'.

5) Enter the first and each succeeding record entry. Each line of input will represent the 32-word record, having only two of the words input by the CRT operator. Enter:

- a) Word one - the time in sequential seconds (integer).
- b) A space to separate entries.
- c) Word two - the alternating "on-off" status code (integer).
- d) A carriage return.

6) To terminate record entries, enter:

- a) Word one - any negative integer value.
- b) A space to separate entries.
- c) Word two - an "off" status code.
- d) A carriage return.

The program will respond with a '**** PAUSE' message.

7) Depress the 'CONTROL' and 'P' keys simultaneously. This will quit the program execution.

8) Close the output file by entering the system command 'C ALL'.

9) If a 'COMO' file was initiated, terminate it by entering 'COMO -END'. To spool the COMO file and get a hard copy at the line printer, enter 'SPOOL filename'.

1b) If a COMO file was not made, a copy of the output file can be obtained by executing the Integer Read (INTRED) Program. (Appendix D.4 is a sample of the INTRED program).

a) Set up a COMO file by entering 'COMO filename' for eventual spooling to the line printer.

b) Before executing the INTRED program, the CRT operator must open the output file for 'reading only.' Enter 'OPEN filename 1 1'.

c) Begin execution by entering 'SEG #INTRED'. The system will respond with a 'GO' and print the file to the CRT until the end of file is reached. Then, the program will respond with a '**** PAUSE' message.

d) Depress the 'CONTROL' and 'P' keys simultaneously. This will "QUIT" the program execution.

e) Close the file by entering the system command 'C ALL'.

f) Terminate the COMO file by entering 'COMO -END'. Spool the COMO file to the line printer with a 'SPOOL filename' command.

APPENDIX D.4 SAMPLE OUTPUT OF THE INTRED PROGRAM

OK, OPEN SAMP1 1 1
OK, SEG #INTRED
GO

0	1
300	2
600	1
900	2
1200	1
1500	2
1800	1
2100	2
2400	1
2700	2
3100	1
3400	2
3700	1
3900	2
4200	1
5000	2
5500	1
6000	2
6500	1
7000	2
7500	1
8000	2
8300	1
12000	2
15000	1
27000	2

NOTE:

OPERATOR INPUT
UNDERLINED.

**** PAUSE

OK, (Depress the CONTROL/P keys simultaneously)
QUIT,
C ALL
OK, COMO -END

ORIGINAL PAGE IS
OF POOR QUALITY

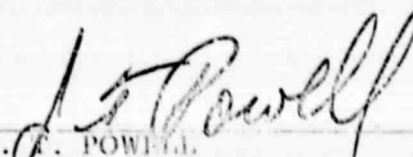
C-2

APPROVAL

SCHEDULER SOFTWARE FOR TRACKING AND DATA RELAY SATELLITE SYSTEM LOADING ANALYSIS - USER'S MANUAL AND PROGRAMMER GUIDE

By Ray Craft, Christel Dunn, Janet McCord,
and Louis Simeone

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.


J. F. POWELL
Chief, Data Systems Laboratory